

ENTWICKLUNG EINES COMPUTERMODELLS
FÜR DEN BIOLOGIEUNTERRICHT
ZUR VERANSCHAULICHUNG
AUSGEWÄHLTER ZUSAMMENHÄNGE
IM ÖKOSYSTEM "SEE"

Dissertation der Fakultät für Biologie
der Ludwig-Maximilians-Universität München

eingereicht am: 7. November 1987

Peter G. Hiering

1988



ENTWICKLUNG EINES COMPUTERMODELLS FÜR DEN BIOLOGIEUNTERRICHT

ZUR VERANSCHAULICHUNG AUSGEWÄHLTER ZUSAMMENHÄNGE

IM ÖKOSYSTEM "SEE"

Inaugural-Dissertation

zur Erlangung des Doktorgrades

der Fakultät für Biologie

der Ludwig-Maximilians-Universität München

vorgelegt von

Peter G. Hiering

aus Unterhaching

November 1987

1. Berichterstatter: Prof. Dr. W. Killermann
2. Berichterstatter: Prof. Dr. O. Siebeck

Tag der mündlichen Prüfung: 25.02.1988

VORWORT

Die nach wie vor schnell steigende Leistungsfähigkeit der Arbeitsplatzrechner (Personal Computer) bietet immer günstigere Voraussetzungen, um auf diesen Geräten leistungsfähige Programme mit hohem Bedienungskomfort einsetzen zu können. Für den Bereich Schule und Ausbildung wird daher der Einsatz der Personal Computer als Hilfsmittel in verschiedenen Schulfächern immer interessanter. Andererseits nehmen aber die Anforderungen an die Entwicklung der erforderlichen leistungsfähigen Programme enorm zu. Die Softwareentwicklung erhält Projektcharakter und bedarf daher einer sorgfältigen Planung und eines systematischen Vorgehens. Für die Fachdidaktik genügt es nicht, sich z.B. nur mit den Kriterien für den unterrichtlichen Einsatz des Computers als Medium oder mit der Beurteilung fertiger Unterrichtsprogramme auseinanderzusetzen; sie darf auf keinen Fall den aktuellen Stand der Software-Entwicklung aus dem Auge verlieren. Die vorliegende Arbeit möchte hierzu einen Beitrag leisten.

Mein Dank gilt besonders Herrn Prof. Dr. Killermann für die Betreuung und Unterstützung der Arbeit sowie die vielen hilfreichen Ratschläge. Danken möchte ich auch meinem Kollegen am Institut, Herrn Dr. Preißler, für die wertvollen Gespräche und Anregungen bei der Modellentwicklung. Bedanken möchte ich mich weiterhin bei Herrn Prof. Dr. O. Siebeck und den Mitarbeitern des Zoologischen Instituts, Herrn Dr. Walz, Herrn Henschel und Frau Kugler für die Ratschläge zu einigen limnologischen Fragen. Dank schulde ich auch den Lehrkräften, Seminarleitern und Schülern, die sich der Mühe der Erprobung und Bewertung des Computermodells unterzogen haben.

INHALTSVERZEICHNIS

1.	EINLEITUNG	S. 6
2.	Didaktische Funktionen der Computersimulation im Biologieunterricht	S. 12
2.1.	Computersimulation als Ersatz für reale Experimente	S. 14
2.2.	Computersimulation als eigenständige Arbeit mit Modellen	S. 15
2.3.	Computersimulation als Mittel der Veranschaulichung	S. 16
3.	Anforderungen an Simulationsprogramme für den Biologieunterricht	S. 19
4.	Verfahren zur Entwicklung von Simulationsprogrammen für den Biologieunterricht	S. 26
5.	Entwicklung des Computermodells "LAGO"	S. 35
5.1.	Zielsetzung	S. 35
5.2.	Charakterisierung des Ökosystems "See"	S. 38
5.3.	Erstellung des Wortmodells	S. 41
5.4.	Erstellung des formalen Modells	S. 46
5.4.1.	Vorüberlegungen	S. 46
5.4.2.	Morphometrische und hydrologische Eigenschaften des Modellsees	S. 54
5.4.3.	Licht	S. 61
5.4.4.	Wassertemperatur	S. 67
5.4.5.	Thermische Schichtung	S. 68
5.4.6.	Phosphatkreislauf	S. 72
5.4.7.	Primärproduktion	S. 82
5.4.8.	Sekundärproduktion	S. 89
5.4.9.	Sauerstoff	S. 92
5.5.	Erstellung des Computerprogramms	S. 96
5.5.1.	Auswahl der Programmiersprache	S. 96
5.5.2.	Erläuterungen zum Computerprogramm	S. 99
5.6.	Darstellung des Modellverhaltens	S. 105

5.7.	Evaluation des Computermodells	S. 110
5.7.1.	Überprüfung der Modellgüte	S. 110
5.7.2.	Erprobung im Unterricht	S. 112
6.	Diskussion	S. 121
7.	Zusammenfassung	S. 125
	LITERATURVERZEICHNIS	S. 127

ANHANG

Anlage 1:	Darstellung der Struktur des Computermodells <i>LAGO</i> unter Berücksichtigung der wichtigsten Systemgrößen und ihrer Wechselwirkungen	S. 133
Anlage 2:	Idealisierte Temperatur-Isoplethen des Modellsees	S. 135
Anlage 3:	Zusammenstellung der Gleichungen und Booleschen Funktionen des mathematischen Modells	S. 137
Anlage 4:	Fragebogen	S. 151
Anlage 5:	Menüs (auf dem Bildschirm angebotene Auswahlmöglichkeiten)	S. 153
Anlage 6:	Farbausdrucke	S. 161
Anlage 7:	Computerprogramm	S. 181
Anlage 8:	Laufzeitanalyse für das Computerprogramm	S. 260
Anlage 9:	Bildschirmfoto	S. 264

1. EINLEITUNG

Ausgangspunkt und Problemstellung

Die rasante Entwicklung auf dem Gebiet neuer Informationstechniken und der Datenverarbeitung ist nicht ohne Auswirkungen auf den Bildungsbereich in der Bundesrepublik geblieben. So hat in den vergangenen fünf Jahren in allen Bundesländern der *Computer*, als Hauptdarsteller der "Neuen Medien", verstärkt Einzug in die allgemein- und berufsbildenden Schulen gehalten. Initiativen auf europäischer ¹⁾ und nationaler Ebene zeigen, daß ein Konsens über die grundsätzliche Bedeutung der neuen Informationstechniken für die Bildung und Ausbildung der Schüler besteht. Für die Bundesrepublik wurden Ziele und Inhalte einer "informationstechnischen Bildung" in einem zwischen dem Bund und den Ländern abgestimmten Rahmenkonzept ²⁾ festgelegt. Dieses Konzept sieht unter anderem vor, daß der Computer nicht nur als Lerngegenstand in Fächern mit unmittelbarem Computerbezug, z.B. Informatik oder Datenverarbeitung, eingesetzt werden soll. Von wesentlicher Bedeutung für die informationstechnische Bildung ist auch der *Anwendungsaspekt*, d.h. der Computer sollte als Hilfsmittel in anderen Fächern, vor allem im mathematischen, naturwissenschaftlichen und wirtschaftlichen Bereich, Verwendung finden.

Zu den Schulfächern, die sich für einen anwendungsorientierten Einsatz des Computers anbieten, zählt zweifellos die Biologie. In zahlreichen Veröffentlichungen wird sogar festgestellt, daß sich mit der Verwendung des Computers als Medium *neue Möglichkeiten* für den Biologieunterricht eröffnen ³⁾. Neben dem Einsatz von Lernprogrammen, der Erfassung und Auswertung von Meßdaten sowie der Speicherung und Verarbeitung umfangreicher Datenbestände steht vor allem die *Computersimulation* im Blickpunkt didaktischen Interesses. An sie werden hohe Erwartungen geknüpft. Viele Fachleute aus Fachdidaktik und Schulpraxis sehen in der Computersimulation ein besonders geeignetes Hilfsmittel, um komplexe Sachverhalte der Biologie durchsichtig und verständlich machen zu können (GÖBEL / KOBLISCHKE 1981, KILLERMANN 1986, SIMON 1980, WEDEKIND 1981, WINDE 1981, 1983, u.a.). Stellvertretend für die allgemein positive Einschätzung sei Dennis MEADOWS, Mitverfasser von "Die Grenzen des

1) vgl. "Entschließung des Rates und der im Rat vereinigten Minister für Bildungswesen" der Europäischen Gemeinschaften vom 19. September 1983

2) vgl. "Rahmenkonzept für die informationstechnische Bildung in Schule und Ausbildung" der Bund-Länder-Kommission für Bildungsplanung und Forschungsförderung (BLK) vom 7. Dezember 1984

3) z.B. KOSCHWITZ 1985, LEHMANN/TISCHER 1987, WINDE 1981

Wachstums" des Club of Rome, zitiert: "Computer und die Modelldarstellung auf Computern können Schülern und Studenten helfen, ein weit umfassenderes Verständnis ökologischer Systeme zu erwerben. Mit Hilfe der Modellierung und der Computersimulation wird es erstmals möglich, im Lehrsaal stellvertretend mit Systemen zu experimentieren, die in der Wirklichkeit sich nur in Jahrzehnten verändern." (aus BOSSEL 1985, Vorwort, S. VII).

Das Zitat von MEADOWS macht auch deutlich, daß man sich von der Computersimulation viel im Hinblick auf die unterrichtliche Vermittlung *ökologischer Themen* verspricht. In der didaktischen Literatur fehlt es nicht an Ausführungen, die für den Ökologieunterricht besondere, in anderen Fächern in dieser Weise nicht vorhandene Probleme sehen. Zwei Beispiele mögen dies verdeutlichen. FELS weist darauf hin, daß eine streng eindimensionale Abfolge von Begriffen, die sonst als besonderes Prädikat eines didaktisch gut aufgebauten Unterrichts gilt, bei vielen ökologischen Sachverhalten nicht zu den erwünschten Einsichten führt. "Ökologische Aussagen kann man nicht auf eine lineare Folge bringen. Ihre Inhalte sind mehrdimensionale Bedingungs Zusammenhänge, die ein Aggregat darstellen ... (FELS 1981, S. 20). Für SCHAEFER gilt als Grundsatz einer Didaktik der Ökologie der von ihm geschaffene Begriff des "Inklusiven Denkens", das durch eine Ausweitung des Objektbereichs auf immer umfassendere Systeme (Ökosysteme, Biosphäre) sowie durch eine Zunahme der Zahl der Sichtweisen gekennzeichnet ist (SCHAEFER 1981, S. 26). Mit den Simulationsmodellen glaubt man, das ideale Hilfsmittel zur Verfügung zu haben, um solchen Besonderheiten besser gerecht werden zu können.

Nun wäre eigentlich zu erwarten, daß die Computersimulation aufgrund der genannten vorteilhaften Eigenschaften in der Unterrichtspraxis eine angemessene Rolle spielen würde. Dies ist aber keineswegs der Fall. Obwohl in den letzten Jahren eine große Zahl von Arbeitsplatzrechnern in den allgemein- und berufsbildenden Schulen Verbreitung gefunden hat, wird im Biologieunterricht nur sehr selten mit Computermodellen gearbeitet. Diese Diskrepanz zwischen den hochgesteckten Erwartungen an das Medium "Computersimulation" und der schulischen Realität macht es erforderlich, noch eingehender, als es bisher geschehen ist, nach den Ursachen zu fragen. Es muß in Betracht gezogen werden, daß die "neuen Möglichkeiten", die man sich von der Computersimulation verspricht, für den Biologieunterricht letztlich nicht von Belang sind. Damit hätte sich das Problem von selbst erledigt. Es kann aber auch

sein, daß wichtige Voraussetzungen fehlen, damit eine eventuell vorhandene spezifische Leistungsfähigkeit überhaupt zur Auswirkung kommen kann. Die primäre Aufgabe besteht also darin, zu klären, welche dieser Voraussetzungen derzeit noch nicht in ausreichendem Maß erfüllt sein könnten und dann ggf. zu versuchen, sie zu realisieren.

Grundlage für einen effizienten Einsatz der Computersimulation sind geeignete Unterrichtsverfahren, ausreichende Kenntnisse im Umgang mit dem Computer bei den Anwendern und geeignete Computerprogramme (Software). Den letzten Punkt möchte ich herausgreifen, da mir hier eine der Hauptursachen für die oben angesprochene Diskrepanz zu liegen scheint. Analysiert man die derzeitige Softwaresituation, so muß festgestellt werden, daß es zwar eine größere Zahl von Simulationsprogrammen gibt, die teils in Fachbüchern und Fachzeitschriften veröffentlicht ⁴⁾, teils von Lehrmittelfirmen angeboten werden ⁵⁾. Bei der Mehrzahl handelt es sich aber um sehr einfache Programme, die, nach Ansicht vieler Anwender, den technischen Aufwand für den Einsatz der Computersimulation im Unterrichtsalltag nicht gerechtfertigt erscheinen lassen. *Es besteht also ein erheblicher Mangel an Programmen, die die speziellen Möglichkeiten der Computersimulation für den Biologieunterricht angemessen nutzen.*

Um diesem Mangel abzuhelpen und damit eine der Grundvoraussetzungen für einen effektiven Einsatz der Computersimulation zu schaffen, ist es erforderlich, sich eingehend mit den Anforderungen an die Qualität der Computerprogramme zu befassen. Eine genaue Kenntnis dieser Zielvorstellungen ist aber noch keineswegs ausreichend. Die Programme müssen auch erstellt werden. Die Entwicklung geeigneter Simulationsprogramme für den Biologieunterricht ist aber extrem zeitaufwendig und stellt hohe Anforderungen an die gleichzeitige Fachkompetenz in sehr unterschiedlichen Bereichen (Biologie, Mathematik, Informatik und natürlich Didaktik). In Frageform gebracht, können die angesprochenen Grundprobleme folgendermaßen umrissen werden:

4) vgl. z.B. MEYER 1984, MEYER/MEYER 1975, REICHAUT 1984

5) vgl. z.B. CVK: Programm 1987, S. 17; WESTERMANN-VERLAG: Biologie auf dem Apple II, Katalog Schulbücher, Klassen 5-13, 1987, S. 46; Carolina Biological Supply Company, Catalog 57, 1986-87, S. 410-413

Frage 1: Welche spezifischen, d.h. durch andere Medien nicht erreichbare Möglichkeiten bietet die Computersimulation für den Biologieunterricht ?

Frage 2: Welche Forderungen müssen an die *Programme* gestellt werden, damit

- a) die spezifischen Vorteile, die man sich vom Medium "Computersimulation" für den Biologieunterricht verspricht, voll zum Tragen kommen können, und
- b) auch solche Lehrer und Schüler mit den Programmen zurechtkommen, die nur selten mit dem Computer arbeiten und keine besondere Ausbildung für seine Bedienung erhalten haben ?

Frage 3: Welches Verfahren ist geeignet, um auf effektive Weise Programme zu entwickeln, die die bei Frage 2 genannten Forderungen erfüllen ?

Ziel der Arbeit

Antworten auf diese Fragen sollen nicht nur auf einer Ebene theoretischer Erörterungen, sondern vor allem anhand eines konkreten Beispiels gegeben werden. Neben einer Auseinandersetzung mit den didaktischen Funktionen der Computersimulation und den Anforderungen an geeignete Simulationsprogramme steht daher die Absicht im Mittelpunkt der Arbeit, ein Computermodell für den Biologieunterricht zu entwickeln, das einige der bei Frage 1) genannten spezifischen Möglichkeiten der Computersimulation beispielhaft zum Tragen bringt, das ferner die Anforderungen an eine hohe Bedienungsfreundlichkeit erfüllt, auf den neueren, an den Schulen verwendeten Computern lauffähig ist und rechnertypische Eigenschaften angemessen nutzt (Frage 2).

Forderungen an geeignete Computerprogramme sind von verschiedenen Autoren und Gremien aufgestellt worden ⁶⁾. Diese Forderungen enthalten Gesichtspunkte, die für die gesamte Palette der Computerprogramme gelten sollen. Zur Entwicklung eines Simulationsprogrammes für den Biologieunterricht sind aber zu-

6) vgl. z.B. BARTH 1985, DIEBOLD 1984, KEIL 1985, KOSCHWITZ 1985, Deutscher Verein zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts 1985

sätzliche, spezifische Kriterien erforderlich. Es ist daher eine weitere Aufgabe dieser Arbeit, diese Kriterien aufzustellen und zu diskutieren.

Bei der Auswahl eines geeigneten Sachverhalts fiel die Entscheidung zugunsten einer synökologischen Fragestellung: Das Computermodell soll ausgewählte Zusammenhänge im Ökosystem "See" veranschaulichen. Diese Thematik ist für eine Computersimulation besonders geeignet⁷⁾. Für die Modellierung bietet das Ökosystem "See" den Vorteil, daß es einen geschlosseneren Charakter aufweist als andere Ökosysteme (z.B. Ökosystem "Wald"). Außerdem sind ausreichend quantitative Daten, teils aus Freiland-, teils aus Laboruntersuchungen vorhanden. Als Einsatzbereich für das Computermodell ist die Oberstufe des Gymnasiums (Sekundarstufe II) vorgesehen. Das Modell erhält den Namen *LAGO*.

Bei der Entwicklung dieses Computermodells für den Unterricht konnte auf keine entsprechenden Erfahrungen anderer Autoren zurückgegriffen werden. Es gibt zwar, wie bereits angemerkt, zahlreiche einfache Modelle, meist zu Themen der Populationsökologie. Sie berücksichtigen aber nur sehr wenige unterschiedliche Faktoren. Typisches Beispiel hierfür sind die auf einem Zwei-Partner-System beruhenden Räuber-Beute-Modelle (z.B. MEYER 1984, SCHOOF 1986, CVK-Computerprogramme 1987 u.v.a). Gleiches gilt für Modelle, die Strukturen und Vorgänge in einem Ökosystem veranschaulichen wollen. Auch hier ist die Zahl der miteinander gekoppelten Systemfaktoren gering (z.B. Modell "POND" in GÖBEL/ KOBLSCHKE 1981). Bisher ist noch kein Versuch unternommen worden, ein Simulationsprogramm für den Biologieunterricht zu entwickeln, das einen überzeugenden Eindruck vom komplexen Zusammenwirken verschiedener biotischer und abiotischer Faktoren und damit von der hohen Vernetzung in Ökosystemen gibt und die genannten Forderungen an Programme, die sich für den Einsatz im Unterrichtsalltag eignen, erfüllt.

Die Entwicklung von Simulationsprogrammen für den Unterricht ist, wie bereits erwähnt, ein sehr zeitaufwendiger Vorgang, bei dem Arbeiten in sehr unterschiedlichen Bereichen geleistet werden müssen. Man kommt auch zu diesem Anwendungszweck nicht ohne ein Mindestmaß an Systematisierung aus, wenn der Entwicklungsprozeß überschaubar bleiben und ein rationelles Vorgehen gewährleistet sein soll. Dazu gibt es bisher keine Veröffentlichungen. Daher besteht ein weiteres Ziel dieser Arbeit darin, ein speziell für didaktische

⁷⁾ vgl. Kapitel 3, S. 19: Anforderungen an Simulationsprogramme für den Biologieunterricht

Zwecke geeignetes Verfahren für die effiziente Entwicklung von Computermodellen vorzustellen und am Beispiel *LAGO* anzuwenden (Frage 3).



2. Didaktische Funktionen der Computersimulation im Biologieunterricht

In den bisherigen Ausführungen war die Rede von den "spezifischen Möglichkeiten", die die Computersimulation für den Biologieunterricht bieten soll und die sie von den traditionellen Medien unterscheiden könnte. Diese Möglichkeiten sind nun konkret zu erläutern. Eine Auseinandersetzung mit den didaktischen Merkmalen und dem didaktischen Stellenwert der Computersimulation ist auch für die Entwicklung des Computermodells *LAGO* von grundlegender Bedeutung.

In den folgenden Abschnitten werden zunächst einige Begriffe geklärt. Dies ist notwendig, weil z.B. der Begriff "Simulation" bei verschiedenen Autoren mit unterschiedlicher Bedeutung verwendet wird ⁸⁾. Im Anschluß daran erfolgt eine kurzgefaßte Darstellung der wesentlichen Gesichtspunkte zu den didaktischen Funktionen der Computersimulation im Biologieunterricht.

Unter "*Simulation*" versteht man "die zielgerichtete Arbeit mit dem Modell eines (realen) Systems" (WEDEKIND 1981, S. 57). Die Simulation setzt also stets die Existenz eines Modells voraus. Dieses hat operativen Charakter, d.h. es ist eine aktive Auseinandersetzung mit ihm beabsichtigt. Die dabei gewonnenen Ergebnisse sind nur für das verwendete Modell gültig, da alle bei der Simulation notwendigen Operationen an diesem Modell und nicht am abgebildeten Realobjekt selbst vollzogen werden.

Zu den bekannten Formen der Simulation zählen das Rollenspiel, das Planspiel und die Computersimulation. WEDEKIND (1981, S. 64) nennt als weitere Simulationsformen die Fallmethode, das Simulationsspiel und das Gedankenexperiment. Bei der Fallmethode wird ein eng begrenzter Realitätsausschnitt in großer Detailtreue abgebildet. Die Teilnehmer müssen ihn dann selbständig analysieren und Lösungsvorschläge für eine klar umrissene Problemstellung ausarbeiten. Simulationsspiele unterscheiden sich von anderen Simulationsformen vor allem durch den im Vordergrund stehenden Spielcharakter.

Bei der *Computersimulation* wird ein Modell in Form eines Simulationsprogrammes auf einen Computer implementiert. Erst der Programmablauf stellt die ei-

⁸⁾ vgl. z.B. BOHRETT/WORDELMANN (1975, S. 22), BMSHOPP/SISSON (1972, S. 25), MANDL (1977, S. 1), LEHMAN (1977, S. 6)

gentliche Simulation dar. Mit dem Computer kann man nur einen Teilbereich der verschiedenen Modelltypen simulieren, nämlich den der *ideellen Modelle* ⁹⁾, die sich als Algorithmen darstellen lassen. Ideelle Modelle drücken die Modelleigenschaften und Relationen in von Menschen definierten Zeichensystemen aus und ermöglichen die Ausführung der gewünschten Modelloperationen in ihnen. Zu diesen Zeichensystemen zählen die natürliche Sprache, bildliche Darstellungen und die mathematische Formelsprache. Von den ideellen Modellen bieten sich für die Umsetzung in ein Computerprogramm besonders die mathematischen und die mit Booleschen Funktionen arbeitenden formal-logischen Modelle an. Bei Verwendung grafikfähiger Rechner kommen auch grafische Modelle in Betracht. Mit fortschreitender Entwicklung der "Künstlichen Intelligenz" und entsprechender Computersprachen (z.B. PROLOG, LISP) werden auch Computermodelle denkbar, die mit dem System der natürlichen Sprache arbeiten.

Von der Computersimulation zu unterscheiden ist die *Computer-unterstützte Simulation*. Bei letzterer wird der Computer nur zur Übernahme bestimmter Teilaufgaben während der Simulation herangezogen. Solche Teilaufgaben können z.B. darin bestehen, daß der Computer zur Durchführung umfangreicher Berechnungen, zum Abruf gespeicherter Informationen oder zur Dokumentation von Zwischen- und Endergebnissen herangezogen wird. Zur Computer-unterstützten Simulation zu rechnen ist auch der Fall, daß der Computer ein physikalisches Modell, z.B. das Modell eines Flugzeugs, für Zwecke der Forschung oder Ausbildung steuert.

Die Begriffsklärung hat deutlich gemacht, daß unter "Computersimulation" immer "das Arbeiten mit dem Modell eines Systems" zu verstehen ist. Ihre Bedeutung ist daher zunächst einmal mit der Bedeutung der Modellmethode für den Biologieunterricht gleichzusetzen. Damit lassen sich für die Computersimulation zwei Funktionsbereiche ableiten, die beide im Biologieunterricht eine Rolle spielen: Sie kann zur Erkenntnisgewinnung oder zur Erkenntnisvermittlung eingesetzt werden. Diese beiden Bereiche sollen nun näher betrachtet werden, mit dem Ziel, festzustellen, welche spezifischen Aufgaben Computermodelle zu übernehmen imstande sind.

9) vgl. Systematik in: KLAUS 1969, S. 414

2.1. Computersimulation als Ersatz für reale Experimente

Im Prozeß der wissenschaftlichen Problemlösung ist die Entwicklung, Überprüfung und Validierung von Modellen, die die untersuchten Phänomene ausreichend beschreiben und einer systematischen Untersuchung zugänglich machen, von zentraler Bedeutung. Auch die Methode der Simulation wird in der Wissenschaft ausgiebig verwendet. In den vergangenen zwei Jahrzehnten hat hier vor allem die Computersimulation eine breite Anwendung gefunden. Sie wird immer dann eingesetzt, wenn die untersuchten Phänomene in einem empirischen, realen Experiment nicht oder nur unter sehr schwierigen Umständen studiert werden können, etwa weil sie zu schnell oder zu langsam ablaufen, zu gefährlich, zu aufwendig, zu komplex oder aus ethischen Gründen nicht durchführbar sind.

Diese heuristische Funktion der Modellmethode und damit der (Computer-) Simulation ist für den Biologieunterricht ebenfalls von Bedeutung, da ein wesentliches Ziel naturwissenschaftlichen Unterrichts darin besteht, dem Lernenden Einblick in die Methoden der wissenschaftlichen Erkenntnisgewinnung zu geben. Ein spezifisches Merkmal der Computersimulation besteht somit in ihrer Funktion als *Experimentierersatz*. Im Biologieunterricht kann es verschiedene Ursachen geben, die ein Experimentieren mit dem originalen Objekt erschweren oder verhindern. Einige typische Beispiele seien hier genannt (vgl. auch WEDEKIND 1981, S. 113):

- Die Zahl der Versuchsobjekte ist zu klein, um bestimmte Gesetzmäßigkeiten zu erfassen (z.B. Vererbungsversuche an Tieren oder Pflanzen).
- Die zu untersuchenden Prozesse laufen zu schnell (z.B. chemische Reaktionen) oder zu langsam ab (z.B. Evolutionsprozesse).
- Das Experiment ist zu gefährlich (z.B. Experimente mit gefährlichen chemischen Substanzen).
- Das Experiment ist zu aufwendig (z.B. Experimente mit einem hohen apparativen Aufwand).
- Das zu untersuchende Objekt ist zu komplex (z.B. ökologische Systeme).
- Das Experiment verbietet sich aus ethischen Gründen (z.B. bestimmte Experimente in der Humanbiologie oder bestimmte Tierversuche).

Die Funktion "Experimentierersatz" gilt auch für die Erkenntnisvermittlung, da mit dem Schulexperiment naturwissenschaftliche Erkenntnisse im strengen Sinn nicht gefunden, sondern veranschaulicht werden (vgl. KILLERMANN 1986, S. 185).

2.2. Computersimulation als eigenständige Arbeit mit Modellen

Bei der Klärung des Begriffs "Computersimulation" wurde deutlich gemacht, daß mit dem Computer nur der Teilbereich der ideellen Modelle simuliert werden kann. Dies ist zunächst eine Einschränkung, andererseits liegen aber gerade hier die besonderen Stärken dieses Mediums. Infolge der spezifischen Fähigkeit des Rechners, im Bereich des mathematischen und formal-logischen Systems komplexe Berechnungen mit hoher Geschwindigkeit auszuführen, können *Modelle mit komplexen Strukturen* entwickelt werden. Solche Computermodelle sind ein spezielles, leistungsfähiges Hilfsmittel, mit dem der Versuch unternommen werden kann, auch komplexe Realsysteme für Zwecke der Forschung oder Lehre zu modellieren. Die Arbeit mit mathematischen Modellen abstrakter Systeme kann dann nicht mehr als "Experimentierersatz" bezeichnet werden, da ein reales Experiment auf dieser Komplexitätsebene nicht durchführbar ist. Man kann in diesem Fall der Computersimulation eine *eigenständige didaktische Funktion* zusprechen, die über das hinausgeht, was mit herkömmlichen Medien möglich ist.

Ein weiterer Anwendungsbereich, der mit "Experimentierersatz" nur unzureichend charakterisiert wäre, ist die simulative Durchführung von *Gedankenexperimenten*, da hierbei Fälle auftreten können, in denen prinzipiell keine realen Experimente möglich sind. Nach KOLLER (1969, S. 54) werden bei Gedankenexperimenten "gedanklich konstruierte Vorstellungen über einen Untersuchungsgegenstand entsprechend den Grundsätzen der experimentellen Methodik variiert und die Konsequenzen dieser Manipulation ebenfalls rein spekulativ abgeleitet ...". Sofern sich die "gedanklich konstruierten Vorstellungen" mathematisch oder formal-logisch beschreiben lassen, ist damit genau das ausgedrückt, was mit den Mitteln der Computersimulation erreichbar ist (siehe Begriffsklärung). Die Affinität zwischen Computersimulation und Gedankenexperiment ist also sehr hoch, weshalb die Computersimulation auch als "materialisiertes Gedankenexperiment" (HARBORDT 1974, S. 52) bezeichnet wird.

"Sieht man einmal von den durch die numerischen Verfahren gegebenen Schritt- und Rundungsfehlern ab, die dazu mit entsprechendem Aufwand nahezu beliebig klein gemacht werden können, bietet die Computersimulation alle Möglichkeiten des Gedankenexperiments mit dem Gewinn zusätzlicher Präzision." (WEDEKIND 1981, S. 138).

Gedankenexperimente spielen in der wissenschaftlichen Forschung für die Theoriebildung, wie sich an vielen historischen Beispielen zeigen läßt, eine große Rolle. Für den Unterricht ist das Nachvollziehen solcher Experimente ebenfalls von Bedeutung, da sich mit ihrer Hilfe in bestimmten Fällen ein besserer Theorieverständnis erreichen läßt. Die Computersimulation ist hierbei besonders dann von Vorteil, wenn sie schwierig zu verstehende Gedankenexperimente durch die Umsetzung in eine "materialisierte" Form leichter zugänglich macht.

2.3. Computersimulation als Mittel der Veranschaulichung

Eine weitere didaktische Funktion der Computersimulation liegt in den speziellen Möglichkeiten der Veranschaulichung von abstrakten Sachverhalten und dynamischen Prozessen. Anschaulichkeit ist eine Grundforderung an jeden Unterricht und deshalb Unterrichtsprinzip. Für den Erkenntnisprozeß besitzt sie herausragende Bedeutung. "Die theoretische Erkenntnis (im Denken) muß verbunden sein mit sinnlicher Erkenntnis (in der Wahrnehmung)" (SIMON 1980, S. 25). Ein anschaulicher Unterricht wird sich bemühen, Beziehungen zwischen den Elementen verschiedenen Abstraktionsgrades herzustellen und die von der Theorie gelieferten Abstraktionen mit (experimentell) nachprüfbar und sinnlich erlebbaren Sachverhalten zu verbinden. Die Computersimulation kann hierzu einen speziellen Beitrag leisten, "weil sie vielfältige Möglichkeiten der Auseinandersetzung mit Modellen erlaubt. Sie erzeugt konkret-situatives Material, über dessen analytisch-synthetische Verarbeitung der Lernende unmittelbar Erfahrungen sammelt, mit dem er zu einem vertieften Verständnis des Problems gelangt, und über das er Intuition für das Systemverhalten entwickelt" (WEDEKIND 1981, S. 131). Die Veranschaulichung kann also durch die Computersimulation auf zweierlei Weise bewirkt werden:

a) durch eine übersichtliche grafische Darstellung, etwa mittels geeigneter Diagramme und Blockstrukturen (Visualisierung)

Während eines Simulationslaufes lassen sich z.B. die zeitlichen Verläufe von Werten sowie die Zustände und Strukturen eines Systems auf dem Bildschirm oder mit Hilfe eines Druckers ausgeben. Grafische Darstellungen können zwar auch mit den herkömmlichen Medien erstellt werden, der Computer kann aber die oft zeitaufwendige Zeichenarbeit übernehmen. Dieser Vorteil macht sich besonders dann bemerkbar, wenn bestimmte grafische Darstellungen nach komplizierten mathematischen Beziehungen, z.B. trigonometrischen Funktionen, Differential- und Integralgleichungen, erstellt werden müssen. Moderne Rechner besitzen in der Regel hochentwickelte Grafikfähigkeiten. Damit könnten für den Unterricht auch die modernen, computerunterstützten Darstellungsmöglichkeiten, die beim Medium Fernsehen inzwischen zum Alltag gehören (in der Fachsprache "Computeranimation" genannt), genutzt werden.

b) durch die aktive Auseinandersetzung mit Modellen, die (simulative) Durchführung von "Experimenten" an diesen Modellen und das schrittweise Sammeln von "Erfahrungen" über die Strukturen und das Verhalten dieser Modelle

Die Computersimulation bietet für den aktiven Umgang mit dem Simulationsmodell eine weitere spezifische Möglichkeit: Der Lernende arbeitet in einem *interaktiven Dialog* mit dem Computer (richtig gesagt, mit dem Programmierer, der den Dialog festgelegt hat). Er kann z.B. Entscheidungen treffen über den Anfangszustand des Modells und kann Parameterwerte bestimmen. Er kann das Verhalten des Computermodells analysieren und daraus Konsequenzen für weitere Simulationsexperimente ziehen. Je flexibler dieser Dialog ist, desto größer ist die Vielfalt der Bearbeitungsmöglichkeiten und damit steigt auch die Intensität der Auseinandersetzung mit dem Modell. HILLE (1980, S. 45) weist darauf hin, daß diese Vielfalt aus denkpsychologischer Sicht von größter Bedeutung ist, da "viele Formen von Denkprozessen und -strategien notwendig [sind], die das gesamte komplexe System der Teiloperationen der analytisch-synthetischen Erkenntnistätigkeit des Menschen ... erfordern."

In obigen Ausführungen ist bewußt vom *Versuch* gesprochen worden, komplexe Realsysteme, die durch eine Vielzahl von Elementen und Wechselbeziehungen

charakterisiert sind, zu modellieren. Auch bei Verwendung der derzeit leistungsfähigsten Computer und Programme zeigen sich jedoch deutliche Grenzen für eine realitätsnahe Simulation biologischer Prozesse. Trotzdem sind Modellbildung und Computersimulation für die wissenschaftliche Forschung von hohem Wert. Gleiches gilt auch für die Verwendung von Simulationsmodellen im Unterricht: Durch die Auseinandersetzung mit dem Modell kann ein besseres Verständnis der Modellzusammenhänge und damit eine vermehrte Einsicht in die Zusammenhänge des realen Systems erreicht werden.

Die Möglichkeit, mit Hilfe der Computersimulation ein besseres Verständnis für komplexe Systeme gewinnen zu können, hat dazu geführt, daß man in diesem Zusammenhang von *neuen Inhalten* ¹⁰⁾ für den Unterricht spricht. Diese Ansicht halte ich nicht für gerechtfertigt. Ein Medium kann keine neuen Inhalte erzeugen, es kann höchstens den Zugang zu bisher verschlossenen oder schwierig erreichbaren Inhalten erleichtern. Was die Computersimulation anbetrifft, so könnten es gerade die in diesem Kapitel aufgezeigten spezifischen Möglichkeiten sein, mit deren Hilfe sich neue Zugänge zu solchen Inhalten finden ließen.

10) vgl. z.B. WEDERKIND 1981, S. 140

3. Anforderungen an Simulationsprogramme für den Biologieunterricht

Die Qualität des Mediums "Computersimulation" ist einerseits von den verwendeten Geräten, andererseits von der Güte der Computerprogramme bestimmt. Eine vergleichbare Abhängigkeit besteht z.B. zwischen Filmprojektor und den vorhandenen Filmen. Damit die im vorigen Kapitel erläuterten didaktischen Funktionen der Computersimulation überhaupt zum Tragen kommen können, sind daher bestimmte Forderungen an die "Tauglichkeit" der Simulationsprogramme für den Biologieunterricht zu stellen. Bei der Festlegung dieser Forderungen sind die Ziele und Inhalte des Schulfaches Biologie, die Unterrichtsmethoden, die Voraussetzungen auf Seiten der Lehrer und Schüler sowie die technischen Möglichkeiten der in der Schule verwendeten Computer in den Blick zu nehmen. Zu den technischen Möglichkeiten ist anzumerken, daß inzwischen die Leistungsmerkmale der neueren, an den Schulen verwendeten Rechner ¹¹⁾ bezüglich Arbeitsgeschwindigkeit, Größe des Arbeitsspeichers und Grafikfähigkeit die mindest erforderlichen Voraussetzungen für eine effiziente Computersimulation im Biologieunterricht erfüllen. Neben allgemeinen Merkmalen, die ein visuelles Medium aufweisen muß, wie z.B. fachliche Korrektheit, Erfüllung allgemeiner didaktischer Grundsätze (motivierender Charakter, Angemessenheit, Anschaulichkeit usw.) und Übereinstimmung der Inhalte mit den Lehrplänen, müssen Computersimulationsprogramme einige zusätzliche Kriterien erfüllen. Diese spezifischen Anforderungen werden zunächst überblicksartig aufgelistet und dann erläutert.

1. Dem Programm soll ein Sachverhalt zugrunde liegen, zu dessen Erarbeitung die Computersimulation besondere Möglichkeiten eröffnet.
2. Das Programm soll interaktives Arbeiten ermöglichen.
3. Das Programm soll anschauliche Grafiken liefern
4. Das Programm soll benutzerfreundlich sein.
5. Das Programm soll auf vielen Schulrechnern lauffähig sein.

11) nähere Erläuterungen zur Ausstattung der Schulen mit Computern siehe S. 23

Dem Programm soll ein Sachverhalt zugrunde liegen, zu dessen Erarbeitung die Computersimulation besondere Möglichkeiten eröffnet.

Wie bereits in der Einleitung der Arbeit bemerkt, bestehen bei der Mehrzahl der Schulpraktiker deutliche Vorbehalte gegen sehr einfache Computerprogramme. Zum Beispiel ist, nach Meinung vieler Biologielehrer, bei den in der Populationsökologie verwendeten einfachen Räuber-Beute-Modelle die traditionelle Arbeit mit Tafel oder Projektionsfolie für das Erstellen der Populationszyklen von vergleichbarer unterrichtlicher Effizienz, aber geringerem Aufwand. Beim Einsatz eines Computers erwarten Lehrer und Schüler eine Hilfestellung, die *deutlich* über das hinausgeht, was auch mit traditionellen Medien möglich ist.

Die spezifische Leistungsfähigkeit der Computersimulation kann aber nur dann zum Tragen kommen, wenn die Computerprogramme die technischen Möglichkeiten der Schulrechner angemessen nutzen. Für die Simulation ist von besonderem Vorteil, daß der Computer mit hoher Geschwindigkeit umfangreiche Datenbestände verarbeitet und zeitaufwendige Berechnungen ausführt. Um diesen Vorteil auszuschöpfen, bedarf es aber ganz bestimmter Sachverhalte, bei deren unterrichtlicher Umsetzung beispielsweise die Auswertung einer großen Zahl von Einzeldaten, umfangreiche mathematische Berechnungen oder eine Vielzahl von Fallunterscheidungen erforderlich ist. Im Fach Biologie herrscht nun gewiß kein Mangel an derartigen Sachverhalten für eine Computersimulation. Paradebeispiel sind ökologische Themen, (z.B. komplexe Nahrungsbeziehungen, Beziehungsgefüge in Ökosystemen), aber auch alle anderen Teilbereiche der Biologie enthalten eine Fülle von Inhalten mit komplexen Strukturen.

Die Forderung nach einer Nutzung der technischen Möglichkeiten ist, nebenbei bemerkt, nicht nur für den Computereinsatz aktuell. Sie gilt für jedes technische Medium. Beispielsweise sind beim Unterrichtsfilm Großaufnahmen, Trickdarstellung, Zeitdehnung und Zeitraffung didaktisch von Vorteil.

Bei der Forderung nach den oben genannten Sachverhalten muß allerdings folgender Fall ausgeschlossen werden: Wenn die im Fach Informatik gewonnenen Kenntnisse und Fähigkeiten im Biologieunterricht angewendet werden sollen, der Schüler also ein Computermodell selbst erstellen oder ein vorgegebenes Programm analysieren soll, muß natürlich ein einfacherer Sachverhalt gewählt werden, da sich der Schüler gleichzeitig mit fachlichen Aspekten, mathemati-

schen Problemen und Fragen des Programmierens auseinandersetzen muß. Diese Art der Arbeit wird im Biologieunterricht eher eine Ausnahme darstellen.

Das Programm soll interaktives Arbeiten ermöglichen.

Auch bei dieser Forderung geht es darum, die zur Verfügung stehenden technischen Möglichkeiten des Computers zu nutzen. Wie bereits in Kapitel 2 erläutert, ist mit dem Rechner ein interaktives Arbeiten möglich, das aus denpsychologischer Sicht von hoher Bedeutung ist (vgl. Zitat von HILLE, S. 17). Im Prinzip können bei einem Computermodell für sämtliche Variablen und Parameter beliebig oft und beliebig viele unterschiedliche Werte eingegeben und ausgewertet und damit alle Aspekte seines strukturellen Zusammenhangs und seines Verhaltens untersucht werden. Zwar wird man in der Regel die gesamte Palette dieser Möglichkeiten auf angemessene Fragestellungen einschränken. Man sollte aber darauf achten, daß die Bearbeitungsmöglichkeiten nicht so weit reduziert werden, daß die Voraussetzungen für selbständiges, forschendes Lernen nicht mehr gegeben sind.

Das Programm soll anschauliche Grafiken liefern.

Bei den meisten moderneren Schulcomputern gehört eine hochauflösende Grafik inzwischen zur Standardausstattung. Damit sind die notwendigen technischen Voraussetzungen gegeben, um die Arbeit mit dem Computermodell durch anschauliche und ansprechende computerunterstützte Grafiken auf dem Bildschirm zu erleichtern. Hierzu zählen z.B. Linien-, Säulen- und Kreisdiagramme sowie dreidimensionale Gitterdarstellungen, die sich während der Simulation trickfilmartig ändern. Solche animierten Grafiken sind sehr gut geeignet, komplexe Modellstrukturen und Zusammenhänge durchsichtig und verständlich zu machen.

Die Computersimulation sollte sich weiterhin die Tatsache zunutze machen, daß Zwischen- und Endergebnisse der Arbeit mit dem Modell mit Hilfe eines Druckers rasch zu Papier gebracht werden können. Im einfachsten Fall liegen diese Ausdrücke in Form von Listen oder Tabellen vor. Es besteht aber auch die Möglichkeit, die grafische Darstellung auf dem Bildschirm zu einem beliebigen Zeitpunkt während der Simulation auszudrucken (Hardcopy). Der didaktische

Vorteil ist vor allem darin zu sehen, daß sich der Zustand eines Computermodells zu einem beliebigen Zeitpunkt festhalten läßt und verschiedenen Ausdrücke vergleichend analysiert werden können.

Das Programm soll benutzerfreundlich sein.

Im Rahmen der Lehrerfortbildung werden derzeit in allen Bundesländern erhebliche Anstrengungen unternommen, um den Lehrern aller Schularten die erforderlichen Kenntnisse im Umgang mit Computern zu vermitteln. Da die Zielgruppe dieser Fortbildung vorwiegend Lehrer für das Fach Informatik umfaßt, können beim Biologielehrer einschlägige Computerkenntnisse nicht als selbstverständlich vorausgesetzt werden. Programme für den Biologieunterricht müssen daher den Mindestanforderungen an die Bedienungsfreundlichkeit genügen. Hierzu zählen vor allem einfache Handhabung (Starten des Programms, Ausdrucken), unmißverständliche Bedienerführung (z.B. in übersichtlichen Menüs ¹²⁾) und Sicherheit gegen Fehlbedienung. Es muß auf jeden Fall vermieden werden, daß der Anwender Kenntnisse über das Betriebssystem eines Computers benötigt, um ein Programm überhaupt starten zu können. Im Idealfall sollte der Vorgang so ablaufen: Der Anwender steckt die Programmdiskette in das Laufwerk und gibt den Programmnamen ein. Alle weiteren Anweisungen erhält er via Bildschirm. Fehleingaben werden ohne unangenehme Nebeneffekte (z.B. Überschreiben der Bildschirmgrafik mit Fehlermeldungen) zurückgewiesen.

Für die Absicherung eines Programms gegen Fehlbedienung ist allerdings ein erheblicher zusätzlicher Zeit- und Arbeitsaufwand bei der Programmerstellung notwendig. Erfahrungsgemäß sind ca. 50% zusätzlicher Programmcode und vertiefte Kenntnisse des Programmierers in der verwendeten Programmiersprache und zum Teil auch auf Betriebssystemebene erforderlich. Trotzdem sollten gerade bei Programmen für den Unterricht die genannten Mindestanforderungen an die Bedienungsfreundlichkeit erfüllt sein.

In diesem Zusammenhang ist noch eine weitere Forderung zu stellen. Sie betrifft die Verfügbarkeit des Programms. Bei vielen Simulationsprogrammen ist nur der Programmcode in Fachbüchern oder Fachzeitschriften veröffentlicht. Möchte der Lehrer mit dem Programm arbeiten, muß er erst das Programm-Li-

¹²⁾ Mit *Menü* wird hier eine am Bildschirm erscheinende Liste von Funktionen bezeichnet, die dem Benutzer zur Auswahl angeboten werden.

sting mühsam abtippen, oft sogar noch zusätzlich an den vorhandenen Rechner anpassen. Der Einsatz von Computermodellen im Unterricht wird so unnötig erschwert. Grundsätzlich sollte die Programmdiskette beim Autor oder Verlag direkt angefordert werden können.

Zur Benutzerfreundlichkeit eines Programms gehört schließlich eine ausreichende Dokumentation. Hierunter ist Begleitmaterial zu verstehen, das auf die erforderlichen Lernvoraussetzungen für die Arbeit mit dem Computermodell eingeht, eine Beschreibung seiner Arbeitsweise und seiner Einsatzmöglichkeiten im Unterricht enthält und, falls nicht im Computerprogramm vorhanden, Hinweise zur Bedienung gibt.

Das Programm soll auf vielen Schulrechnern lauffähig sein.

Zur Erläuterung dieser Forderung ist es zunächst notwendig, auf die derzeitige Ausstattung der Schulen mit Computern einzugehen. Dies ist nicht ganz einfach, denn vielfältige Beschaffungsinitiativen haben zu einem in den verschiedenen Bundesländern mehr oder weniger stark ausgeprägten "Computerwildwuchs" geführt. In den vergangenen Jahren sind z.B. viele unterschiedliche Geräte, die zur Klasse der sogenannten "Homecomputer" ¹³⁾ gezählt werden, angeschafft worden. Im Vergleich zum heutigen Standard sind bei diesen Geräten Arbeitsgeschwindigkeit, Kapazität des Arbeitsspeichers und Grafikfähigkeit gering. Etwa ab 1986 haben leistungsfähigere Computer, in der Arbeit bisher als "neuere an den Schulen verwendete Rechner" bezeichnet, verstärkt Eingang in die Schulen gefunden ¹⁴⁾. Ihr Anteil dürfte derzeit bei etwa 40% liegen. Aber auch diese Geräteklasse ist nicht einheitlich. Unterschiede können z.B. im Diskettenformat, in der Grafikauflösung oder im verwendeten Betriebssystem vorhanden sein. Einen einheitlichen "Schulcomputer", den man z.B. in England versucht hat, einzuführen, gibt es in bundesdeutschen Schulen nicht.

13) Diese Geräteklasse war vor einigen Jahren in der Regel durch einen 8-Bit-Datenbus und die Mikroprozessoren 6502, 6800, 6809, 8080, 8085, 9980/9981 oder Z80 gekennzeichnet.

14) Hier sind Computer gemeint, die als "Arbeitsplatzrechner" (engl.: Personal Computer) bezeichnet werden. Teils gehört diese Geräteklasse zur Gruppe der sog. "IBM"-kompatiblen Rechner mit dem Mikroprozessor 8088 bzw. 8086 und dem Betriebssystem MS-DOS, teils ist sie durch einzelne Geräte auf der Basis des Mikroprozessors 68000 vertreten.

(* IBM ist eingetragenes Warenzeichen der International Business Machines Corporation, USA; MS ist Kürzel für die Microsoft GmbH, Washington, USA)

Die Forderung, ein Programm soll auf einer möglichst großen Zahl unterschiedlicher Rechner eingesetzt werden können, klingt vernünftig, ist aber auf Grund der geschilderten Hardwaresituation nur schwer zu erfüllen. Da die verschiedenen Rechnermodelle untereinander nicht kompatibel sind, ergibt sich die Schwierigkeit, daß ein Computerprogramm in der Regel nur auf dem Rechner lauffähig ist, für den es entwickelt worden ist. Eine Übertragung auf andere Gerätetypen ist entweder überhaupt nicht möglich, z.B. infolge mangelnder Kompatibilität auf Betriebssystemebene, oder aber mit einem sehr hohen Zeitaufwand für die Umarbeitung verbunden ¹⁵⁾. Allerdings bahnt sich hier in letzter Zeit eine deutliche Verbesserung an. Die zuständigen Ministerien sind bestrebt, mit Hilfe entsprechender Empfehlungen die Hardware an den Schulen zu vereinheitlichen, um eine breitere Basis für den unterrichtlichen Einsatz eines geeigneten Programms zu schaffen. Dies wird aber nicht so weit gehen, daß an allen Schulen nur noch Computer eines einzigen Herstellers angeschafft werden. Nebenbei bemerkt wäre selbst in diesem Fall das Kompatibilitätsproblem nicht gelöst, da, wie sich bisher gezeigt hat, die Computerhersteller sich in der Regel wenig um die Kompatibilität ihrer Modelle kümmern. Man kann der Forderung, ein Programm soll auf möglichst vielen Rechnern eingesetzt werden können, jedenfalls auf der Ebene der betriebssystemkompatiblen Computer deutlich näher kommen, wenn bei der Programmierung der Zugriff auf hardware-spezifische Besonderheiten unterbleibt ¹⁶⁾.

Die Ausführungen zu den Anforderungen an Simulationsprogramme für den Biologieunterricht zeigen, daß einige der genannten Forderungen eng an die Leistungsfähigkeit der verwendeten Computer geknüpft sind. Die Computersimulation kann daher erst mit den verbesserten technischen Möglichkeiten der neueren Geräte zu einem geeigneten Hilfsmittel für den Biologieunterricht werden. Der derzeitige Mangel an geeigneten Simulationsprogrammen ist daher auch darin zu sehen, daß diese Computer den Schulen erst seit relativ kurzer Zeit zur Verfügung stehen. Mit der Kritik an der gegenwärtigen Programmsituation

15) Wenn z.B. zwar die verwendete Programmiersprache beibehalten, aber ein anderer Compiler mit einem abweichenden Befehlsumfang verwendet werden muß, kann u.U. der Fall eintreten, daß einige Algorithmen nicht mehr in der bisherigen Weise formuliert werden können. Oft ist damit praktisch eine Neuentwicklung des Programms erforderlich.

16) Gemeint ist hier z.B. die Verwendung der im ROM (Read-Only-Memory) gespeicherten Ein-/Ausgaberroutinen (Basic-Input/Output-System; BIOS) oder das direkte Schreiben in den Bildschirmspeicher.

soll deswegen auch keinesfalls die Arbeit der Programmautoren geschmälert werden, im Gegenteil, die Entwicklung einfacherer Programme unter starken hardwarebedingten Beschränkungen ist mindestens genauso arbeitsintensiv wie heute die Entwicklung komplexerer Software in einer komfortablen Entwicklungsumgebung.

4. Verfahren zur Entwicklung von Simulationsprogrammen für den Biologieunterricht

Bei der Entwicklung eines Computermodells für den Biologieunterricht sind Arbeiten in mehreren, teilweise sehr unterschiedlichen Bereichen erforderlich. Es ist eine Auseinandersetzung mit didaktischen Fragestellungen notwendig. Die biologischen Sachverhalte müssen eingehend analysiert werden. Weiterhin sind die Schwierigkeiten bei der Erstellung eines mathematischen Modells zu lösen. Und schließlich muß auch noch ein Computerprogramm erstellt werden. Für die Entwicklung komplexerer Modelle erweist es sich daher als notwendig, ein Verfahren aufzustellen, das die in jedem der genannten Teilbereiche zu leistenden Aufgaben definiert und die Bearbeitungsschritte in eine sinnvolle Reihenfolge bringt. Außerdem muß bei der Entwicklung von Computermodellen für den Biologieunterricht davon ausgegangen werden, daß die Programmierarbeit in der Regel nicht von professionellen Fachleuten erledigt wird, da hierfür der Markt für Schulprogramme in Relation zu den hohen Entwicklungskosten zu klein ist und die Software zu sehr hohen Preisen angeboten werden müßte. Es muß daher ein Weg gefunden werden, wie der zu erwartende hohe Zeitaufwand bei der Erstellung des Programmcodes durch Nutzung von Hilfsmitteln in ein für den nicht-professionellen Programmierer leistbares Maß gebracht werden kann.

Bearbeitungsschritte bei der Modellerstellung

Ausgangspunkt der Modellentwicklung für den Unterricht sind die aktuellen wissenschaftlichen Erkenntnisse über den zu modellierenden Sachverhalt. Sie liegen teils als theoretische Vorstellungen, teils als experimentelle (empirische) Befunde vor. Anhand dieser Ergebnisse können Struktur und Eigengesetzlichkeit des Sachverhalts erfaßt werden. Diese Auseinandersetzung, im didaktischen Sprachgebrauch "*Sachanalyse*" genannt, schärft den Blick für falsche Aussagen, ungerechtfertigte Verallgemeinerungen und unangemessene Vereinfachungen (vgl. KILLERMANN 1986, S. 238). Hand in Hand mit der Sachanalyse sind grundsätzliche didaktische Überlegungen (*didaktische Analyse*) anzustellen. Hierbei wird nach dem Bildungsgehalt, dem Sinn- und Sachzusammenhang, dem Motivationsgehalt, der Zugänglichkeit usw. eines Lerngegenstands

gefragt (vgl. KLAFFKI 1964). Durch die Bearbeitung solcher Analysierungsschwerpunkte kristallisiert sich die didaktische Zielsetzung des Modells heraus. Ein weiterer didaktischer Bearbeitungsschritt besteht in der **didaktischen Reduktion**. Hierbei werden die fachwissenschaftlichen Sachverhalte auf die Lernvoraussetzungen und Lernfähigkeiten des Lernenden sowie auf die zur Verfügung stehende Unterrichtszeit abgestimmt. Komplexe Inhalte werden auf wesentliche Teile bzw. Zusammenhänge beschränkt und so vereinfacht, daß Verständlichkeit erzielt und gleichzeitig die sachliche Richtigkeit gewahrt wird. Der Sachverhalt läßt sich vereinfachen durch gezielte Ausschnittbildung (sektorale Reduktion) und/oder indem die Struktur hinsichtlich Kompliziertheit und Umfang reduziert wird (strukturelle Reduktion) ¹⁷⁾.

Da das "Werkzeug Sprache" für die Beschreibung von komplizierten Strukturen und Verhaltensweisen von allen zur Verfügung stehenden Hilfsmitteln am leistungsfähigsten ist, werden im nächsten Modellierungsschritt die Eigenschaften des Computermodells, die sich durch die genannte fachwissenschaftliche und didaktische Bearbeitung herauskristallisiert haben, in einem **Wortmodell** fixiert. Aus dem Wortmodell sollen die didaktischen Zielsetzungen sowie Art und Grad der didaktischen Reduktion der Inhalte erkennbar sein. Anzustreben ist eine möglichst detaillierte Beschreibung des erwünschten Modellverhaltens, da das Wortmodell während der Modellentwicklung als Korrektiv dient.

Trotz der vorgenommenen Vereinfachungen ist das Wortmodell – vorausgesetzt es handelt sich nicht um einen sehr einfachen Sachverhalt – für die Entwicklung eines Computermodells noch zu komplex. In einem weiteren Schritt muß versucht werden, die Vielzahl von Einzelercheinungen mittels weniger grundlegender Gesetzmäßigkeiten zu beschreiben. Dies ist durch eine **Formalisierung** möglich. Für die Erstellung eines formalen Modells stehen zwei verschiedene Methoden zur Wahl: (Die folgenden Überlegungen lehnen sich eng an den Text des Lehrbuches von STRASKRABA/GNAUCK 1985, S. 31-64 an.)

a) Stochastische Modellierung (experimentelle statische Modellbildung)

Diese Art der Modellierung berücksichtigt den stochastischen Charakter der zeitlichen und räumlichen Änderungen der Eingangs- und Zustandsgrößen sowie deren Anfangsunbestimmtheit. In der Regel sind für die im System oder Sub-

17) zu den Formen der didaktischen Reduktion vgl. KILLERMAN 1986, S. 241

system ablaufenden zufallsabhängigen Prozesse keine Differentialgleichungen bekannt. Die stochastische Modellierung arbeitet deshalb mit den Methoden der Wahrscheinlichkeitsrechnung und der mathematischen Statistik.

Bei den *statischen Verfahren* wird die Größe "Zeit" nicht in die Systemanalyse einbezogen. Es stehen zur Verfügung: "einfache und mehrfache lineare und nicht lineare Korrelation und Regression, Varianz-, Diskriminanz- und Faktorenanalyse sowie Schätzverfahren ... Eine Sonderstellung innerhalb der Schätzverfahren bilden die indirekten Verfahren, die die Größe "Zeit" durch die Art der angewendeten Schätzprozedur in die Auswertung einbeziehen.

Dynamische Verfahren berücksichtigen die Größe Zeit (Zeitreihenanalyse). Man arbeitet mit Fourier-Analyse, Korrelations- und Spektralanalyse, Bestimmung der Gewichts- und Übertragungsfunktion." (STRASKRABA/GNAUCK 1985, S. 31)

Ein Teilgebiet der stochastischen Modellierung ist ferner die *adaptive Modellierung*. Sie verdient auf Grund der verwendeten selbstanpassenden Algorithmen besondere Beachtung, weil mit dieser Methode die Selbstadaption eines Ökosystems dargestellt werden kann. Als adaptive Steuerung kann beispielsweise die Änderung der Filtriergeschwindigkeit des Zooplanktons bei unterschiedlichen Phytoplanktonmengen verstanden werden. Die Anwendung adaptiver Verfahren ermöglicht eine Systemanalyse, bei der die Anfangsunbestimmtheit durch Ausnutzung der im Prozeßverlauf gewonnenen Informationen verringert wird. Die Realitätsnähe eines Modells kann damit deutlich verbessert werden.

b) Analytische Modellierung (theoretische Modellbildung)

Bei der analytischen Modellierung werden die wesentlichen Eingangsgrößen und die Zustandsgleichungen als bekannt vorausgesetzt. Die vollständige Bestimmtheit gestattet die Anwendung klassischer mathematisch-analytischer Methoden. An Werkzeugen für die mathematische Modellierung stehen die folgenden drei Hauptformen zur Wahl:

■ Mengenlehre und Abbildungen

Sie werden für die Darstellung allgemeiner Modelle benutzt, etwa wenn man den Übergang von gegebenen Zuständen in andere abbilden möchte. Hierbei stellt man eine Liste der Eigenschaften oder Zustände auf, die ein System

annehmen kann und das Modell besteht aus einer Abbildungsregel, die den Status ergibt, den das System nach dem derzeitigen Zustand annehmen wird.

■ Matrixalgebra

Sie dient der Beschreibung und Behandlung von Listen und Zahlentabellen. Matrizes erlauben eine allgemeine symbolische Beschreibung von Systembeziehungen. Die Techniken der Matrixalgebra sind der Grundstein vieler Modelle.

■ Differenzengleichungen, Differentialgleichungen

Sie werden zur Konstruktion von Modellen benutzt, die die zeitliche Änderung eines Systems quantitativ beschreiben.

Bei der Modellierung wird häufig mit Differentialgleichungen gearbeitet, die die Ursache-Wirkungs-Beziehungen in einem Ökosystem beschreiben (Zustandsraummodelle). Die analytische Modellierung geht dann in der Regel in vier Stufen vor sich. "An die erste Stufe, die Konzipierung des Modells und die Aufstellung der Differentialgleichungen, schließen sich die Schritte Parameterschätzung (Parametrisierung), Simulation und Modelltestung an." (STRASKRABA/GNAUCK 1985, S. 47) Ziel der Parametrisierung ist die Bestimmung der quantitativen Werte der Parameter. Der Begriff "Simulation" hat in diesem Zusammenhang eine andere Bedeutung als bei der in Kapitel 2, S. 12 vorgenommenen Begriffsklärung. Da für die Modellierung natürlicher Systeme in der Regel komplexe mathematische Modelle erforderlich sind, bei denen zwischen den einzelnen Komponenten vielfältige, oft nicht lineare Interaktionen ablaufen, ist eine exakte Lösung der hierbei verwendeten Gleichungen oder Gleichungssysteme meist nicht möglich. Analytisch nicht lösbare Differentialgleichungssysteme können dann, unter Zuhilfenahme eines Digitalrechners, durch "Simulation" gelöst werden. Simulation bedeutet hier also "die numerische Berechnung des Verhaltens eines (Gleichungs-)Systems, meist über die Zeit, wobei verschiedene Randbedingungen durchgerechnet werden." (SEITZ 1985, S. 4)

Wie jedes Modell stellt auch ein mathematisches Modell eine Vereinfachung der objektiven Realität dar. Die Verfahren der mathematischen Modellierung sind zwar sehr leistungsfähig, eine deckungsgleiche Abbildung natürlicher Systeme ist aber auch mit ihnen nicht möglich. Bei der Umsetzung des Wortmodells in ein formales Modell müssen, sofern das Wortmodell nicht zu einfach ist, Verluste hingenommen werden. Die Formalisierung führt gegenüber dem Wortmodell

zu einer inhaltlichen Minderung, d.h. Modellbeschreibung und tatsächliches Modellverhalten weichen in einem gewissen Umfang voneinander ab.

Das formale Modell muß im nächsten Modellierungsschritt in eine Form gebracht werden, die der Computer verarbeiten kann: das *Computerprogramm*. Hierzu steht eine Palette von Programmiersprachen, von der Maschinensprache über Assembler bis zu den höheren Programmiersprachen, zur Verfügung. Für die Entwicklung von Simulationsprogrammen gibt es außerdem spezielle "Simulationssprachen". Bei der Programmierung führt die häufig notwendige numerische Lösung analytischer Verfahren zu weiteren "Verlusten" in Form von Approximations- und Rundungsfehlern. Zusammen mit den durch die Formalisierung bedingten Minderungen können sich deutliche, nicht erwünschte Abweichungen im Verhalten des Computermodells gegenüber der Verhaltensbeschreibung im Wortmodell ergeben, die die Computersimulation unter Umständen didaktisch wertlos machen.

Wenn das Computermodell als fertiges, auf den vorgesehenen Rechnern lauffähiges Computerprogramm vorliegt, ist es daher erforderlich, die *Modellgüte* zu überprüfen. Bei der Testung von Modellen, die z.B. zu Zwecken der Ökosystemforschung entwickelt werden, unterscheidet man hierbei zwischen Verifikation und Validation.

"Bei der *Verifikation* werden die Ergebnisse eines Simulationslaufes mit den am System beobachteten *Werten* verglichen. Der Vergleich kann qualitativ oder quantitativ erfolgen. Beim qualitativen Verfahren schätzt man die Übereinstimmung rein subjektiv ein. Quantitative Vergleiche sind objektiver, durch die notwendige Entscheidung für ein Vergleichskriterium ist aber auch hier eine gewisse Subjektivität enthalten." (STRASKRABA/GNAUCK 1985, S. 59)

Bei der *Validation* (auch Validierung genannt) vergleicht man die *Reaktionen* und die *Dynamik* des Modells (Gesamtmodell oder Modellteile) mit dem entsprechenden Verhalten des realen Objekts. Eine von STRASKRABA/GNAUCK vorgeschlagene, leistungsfähige Validationsmethode besteht darin, mit Hilfe einer Serie von Simulationen die Modellantwort auf systematisch geänderte Umgebungs- oder Störgrößen zu erfassen. Voraussetzung für den Vergleich mit dem Realsystem ist allerdings, daß die zugrundeliegenden Abhängigkeiten in diesem System bekannt sind (vgl. STRASKRABA/GNAUCK 1985, S. 60).

Zur Überprüfung der Modellgüte von Computermodellen, die für didaktische Zwecke entwickelt werden, sind ebenfalls beide Arten der Modelltestung, allerdings mit zum Teil veränderter Aufgabenstellung und Wertigkeit, heranzuziehen. Durch die Verifikation soll sichergestellt werden, daß die Werte, die der Benutzer beobachten kann oder die er aktiv verändert, an der Realität orientiert sind. Es wird bewußt die Vokabel "orientiert" verwendet. Da ein didaktisches Modell nicht einem bestimmten, real existierenden Objekt entsprechen, sondern allgemeingültigen Charakter haben soll, ergibt sich folgende Schwierigkeit: Beim Vergleich mit den in der Literatur veröffentlichten oder selbst gemessenen Werten, die eben nur für ein bestimmtes Realobjekt gelten, ist immer eine Interpretation erforderlich, welche Werte für generalisierende Aussagen brauchbar sind bzw. muß ein für eine größere Bandbreite gültiger Mittelwert abgeleitet werden. Die Verifikation ist daher stark subjektiv; sie darf aber dennoch nicht übergangen werden, wenn das Modell Systemvariablen quantitativ darstellt.

Die Validation unterscheidet sich von dem in der wissenschaftlichen Forschung üblichen Vorgehen vor allem in folgendem Punkt: Vorrangiges Korrektiv ist nicht das Realsystem, sondern das im *Wortmodell* festgelegte Modellverhalten. In der Regel ergibt sich durch die didaktische Reduktion und durch das Bestreben, allgemeingültige Aussagen über den zu simulierenden Sachverhalt zu finden, ein idealisiertes Modell mit qualitativem Charakter. Eine direkte Entsprechung zu einem bestimmten, real existierenden Objekt ist dann meist nicht mehr möglich und auch nicht beabsichtigt. Es wäre aber falsch, ein idealisiertes Modell als realitätsfremd zu bezeichnen, da es sich bei sorgfältiger Ausarbeitung stets an den vorhandenen wissenschaftlichen Erkenntnissen über den realen Sachverhalt orientiert.

Im Anschluß an die Überprüfung der Modellgüte ist bei Simulationsmodellen für didaktische Zwecke eine weitere Testphase erforderlich, die Erprobung des Computermodells im Unterricht. Hierbei soll festgestellt werden, inwieweit das Modell - je nach Zielsetzung - eine oder mehrere der möglichen didaktischen Funktionen (vgl. Kapitel 2) erfüllt und ob es den Anforderungen an ein unterrichtstaugliches Computerprogramm (vgl. Kapitel 3) genügt. Während die bei einer Optimierung der Modellgüte erforderlichen Änderungen in erster Linie das formale Modell und das Computerprogramm betreffen, kann sich durch die

bei der unterrichtlichen Erprobung festgestellten Mängel auch eine Überarbeitung des Wortmodells als notwendig erweisen. Die Überprüfung der Modellgüte und die Erprobung des Modells im Unterricht werden bei dem hier vorgestellten Verfahren als *Evaluation* bezeichnet. Als endgültig abgeschlossen kann die Evaluation erst nach einem längerfristigen Einsatz im Unterrichtsalltag gelten.

Zusammenfassend soll die folgende Abbildung die Abfolge der verschiedenen Bearbeitungsschritte bei der Modellentwicklung für didaktische Zwecke, die sich wesentlich von der Modellentwicklung für Anliegen der Ökosystemforschung unterscheiden (vgl. Fig. 1.1. in STRASKRABA/GNAUCK 1985, S.13), grafisch veranschaulichen:

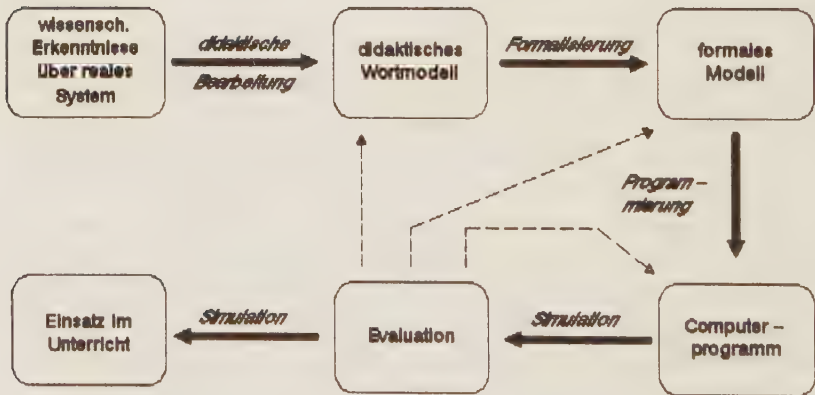


Abbildung 1: Vorgehensweise bei der Entwicklung eines Computermodells für den Unterricht

Programmierhilfen

Für die Geräteklasse der neueren Arbeitsplatzrechner (vgl. Anmerkung in Fußnote 14, S. 23) stehen inzwischen für jede Programmiersprache leistungsfähige Programmierhilfen (sogenannte "Tools") zur Verfügung ¹⁹⁾. Jedes dieser Tools ist ein professionelles Programm, dessen Aufgabe darin besteht, die Entwicklung komplexerer Programme zu erleichtern und zu beschleunigen. Diese Programmierhilfen können daher wesentlich dazu beitragen, den hohen Zeitbedarf bei der Entwicklung von Simulationsprogrammen zu vermindern.

Bei den Programmierhilfen unterscheidet man "Front-End-Tools" und "Back-End-Tools". Erstere werden zur Ausarbeitung des Programmcodes – also vor dem Kompilieren ¹⁹⁾ – eingesetzt, letztere werden verwendet, wenn ein Programm beim Probelauf Fehler aufweist. Zu den Front-End-Tools gehören z.B. komfortable, bildschirmorientierte Editoren ²⁰⁾ und Syntax-Checker ²¹⁾, mit deren Hilfe sich auf effektive Weise ein Programmcode ohne syntaktische Fehler erstellen läßt. Ein wichtiges Werkzeug bei den Back-End-Tools ist der Debugger ²²⁾. Bei Verwendung eines leistungsfähigen Debuggers kann enorm viel Entwicklungszeit eingespart werden, da sich selbst knifflige Laufzeitfehler, wie z.B. das Überschreiten eines zulässigen Wertebereichs, was etwa bei Funktionen mit exponentiellem Verlauf schnell der Fall sein kann, durch eine gezielte Analyse der Fehlerumgebung relativ rasch erkennen lassen. Weitere während des Probelaufs dienliche Hilfsprogramme sind *Linker* ²³⁾, *Make*-Programme ²⁴⁾ und Programme zur Laufzeitorientierung des Programmcodes.

Weitere Hilfsprogramme zur Programmentwicklung sind sogenannte "Bibliotheken". Darunter versteht man eine Sammlung fertiger Routinen (Programmteile), die auf einfache Weise in ein zu entwickelndes Programm eingefügt werden können. In käuflichen Programmsammlungen finden sich eine große Zahl ferti-

18) vgl. Marktübersicht in Computer Persönlich. Das aktuelle Fachmagazin für Personal Computer, Haar b. München, Heft 14, 1987

19) *Kompilierer* (Übersetzer): Ein Kompilierer ist ein Programm, das den in einer höheren Computersprache erstellten Programmcodes in einen Code übersetzt, der vom Computer direkt verarbeitet werden kann (den sogenannten Maschinencode).

20) Ein *Editor* ist ein Programm, mit dessen Hilfe der Programmcodes per Tastatur eingegeben und bearbeitet werden kann.

21) Ein *Syntax-Checker* ist ein Programm, das Verstöße gegen die syntaktischen Regeln im Programmcodes findet und anzeigt.

22) Ein *Debugger* (Debug Programm) ist ein Programm, das die Suche von logischen Fehlern unterstützt. Die wichtigste Funktion eines Debuggers besteht darin, daß während des Testlaufs ein beliebiger Abbruchpunkt gesetzt werden kann, an dem dann z.B. die Werte der Variablen, die Speicher- und die Registeradressen überprüft werden können.

23) *Linker* (Binder): Systemprogramm zur Verbindung von einzelnen Programmsegmenten zu einem Programm, das direkt ausgeführt werden kann.

24) *MAKE*-Programm: Programm, das nur die Module (Teilprogramme) neu kompiliert, die verändert worden sind. Damit läßt sich unnötige Wartezeit während der Kompilierung umfangreicher Programme einsparen.

ger Lösungen zu schwierigeren Aufgaben, wie z.B. die Ausgabe von grafischen Darstellungen auf dem Bildschirm, das Generieren von Bildschirmmasken, das Erstellen einer Dateiverwaltung mit umfangreichen Datenbeständen, die Ausführung komplizierterer mathematischer Berechnungen (Matrizen- und Vektorenrechnung, Fouriertransformation und -entwicklung, lineare Randwertproblem, komplexe Arithmetik ...) und viele mehr. Die Routinen sind allerdings in den seltensten Fällen auf das zu lösende Problem "maßgeschneidert", da bei ihrer Entwicklung ein universeller Anwendungsbereich angestrebt wird; dieser Nachteil wird aber durch die oft erhebliche Zeitersparnis beim Programmieren mehr als wettgemacht.

5. Entwicklung des Computermodells *LAGO*

5.1 Zielsetzung

Ziel der Entwicklung des Computermodells *LAGO* ist es, das Zusammenwirken abiotischer und biotischer Faktoren im Ökosystem "See" deutlich zu machen. Aus der Fülle der für dieses System charakteristischen Eigenschaften müssen einige wesentliche ausgewählt werden, um es transparent und verständlich machen zu können. Die Reduktion darf aber nicht zu weit gehen. Bei der Modellierung eines Ökosystems muß die Schwierigkeit gemeistert werden, daß einerseits das System durch didaktische Vereinfachung durchschaubar gemacht werden soll, aber andererseits der Eindruck der Komplexität als wesentliches Merkmal auf jeden Fall erhalten bleiben muß. Genaue Ausführungen, welche Systemvariablen berücksichtigt werden sollen und welches Modellverhalten angestrebt wird, erfolgen bei Punkt 5.3. (S. 41, "Wortmodell").

Mit dieser Zielsetzung wird für die Computersimulation die didaktische Funktion der "eigenständigen Arbeit mit Modellen" (vgl. Abschnitt 2.2., S. 15) angestrebt. Damit auch die erwünschte Funktion der Veranschaulichung (vgl. Abschnitt 2.3., S. 16) realisiert werden kann, sollen hierfür die spezifischen Möglichkeiten des Computers genutzt werden. Der zeitliche Verlauf der ausgewählten Zustandsgrößen soll mit Hilfe anschaulicher, möglichst selbsterklärender Grafiken auf dem Bildschirm dargestellt und während der Simulation zu jedem beliebigen Zeitpunkt auf einem Drucker ausgegeben werden können. Die gesamte Darstellung soll übersichtlich, aber doch so reichhaltig sein, daß deutlich wird, welche Fülle von Änderungen sich pro Zeitschritt im Modellökosystem vollzieht. Weiterhin ist beabsichtigt, dem Anwender viele Möglichkeiten des interaktiven Arbeitens mit dem Computermodell zu geben, damit er durch eine Vielzahl von Experimenten Erfahrungen über das Verhalten des Modells gewinnen kann. Die hierzu notwendige Änderung von Parametern soll über leicht verständliche und leicht zu bedienende Auswahlmenüs erfolgen. Eventuelle Fehleingaben (falsche Werte, Bereichsüberschreitungen) müssen zurückgewiesen werden, ohne daß eine Störung des Programmablaufs eintritt.

Eine unverzichtbare Bedingung, damit das Computermodell zur Unterstützung und Ergänzung des Biologieunterrichts eingesetzt werden kann, ist die Lehr-

planentsprechung. Bei der Durchsicht der Lehrpläne zeigt sich, daß in allen Bundesländern für die Oberstufe der Gymnasien der Themenbereich "Ökosystem" vorgesehen ist. Stellvertretend für die ähnlich lautenden Zielsetzungen sei der Lehrplan von Baden-Württemberg genannt: "Wirkungen von Umweltfaktoren auf Organismen und Kenntnisse über deren gegenseitige Abhängigkeit ermöglichen dem Schüler Einsichten in größere ökologische Zusammenhänge. Auf dieser Grundlage soll er eine verantwortungsvolle Einstellung zur Natur entwickeln, angemessen handeln und mögliche Folgen von Eingriffen in ein Ökosystem abschätzen können." ²⁵⁾ Auch die in den Lehrplänen vorgesehenen Inhalte sind nahezu identisch: Sie enthalten übereinstimmend die Themen Nahrungsnetz, Stoffkreislauf und Energiefluß, Biologisches Gleichgewicht, Struktur eines Ökosystems, Umweltbelastung und Schutzmaßnahmen. ²⁶⁾ An welchem Ökosystem diese Inhalte exemplarisch erarbeitet werden, bleibt dem Lehrer überlassen.

Im Lehrplan von Bayern wird dagegen das Ökosystem "See" konkret genannt: Im Ausbildungsabschnitt 12/2 für Grundkurs und Leistungskurs, 12. Jahrgangsstufe, ist unter anderem der Themenbereich "Ökologie und Umweltschutz" vorgesehen ²⁷⁾. Die Schüler sollen am Beispiel "See" dieses Ökosystem als ein "Fließgleichgewicht" verstehen, das sich durch die "Ausgewogenheit der gegenseitigen Beziehungen" erhält (Lernziel 5.2, Leistungskurs bzw. Lernziel 2.3, Grundkurs). An Lerninhalten sind hierzu genannt:

- Gliederung des Sees in verschiedene Lebensräume
- Thermische Gliederung und Wasserzirkulation
- Bedeutung des Phytoplanktons
- Energiefluß und Stoffkreislauf im Ökosystem: Nahrungsnetz und Nahrungskette (Produzenten, Konsumenten)
- Mineralisierung abgestorbener Organismen
- Bedeutung der Destruenten

25) Baden-Württemberg: Lehrplan Gymnasium. Bildungsplan für das Gymnasium der Normalform, Band 1, Lehrplanheft 8/1984, Villingen-Schwenningen, 1984, S. 926

26) vgl. z.B. Baden-Württemberg, ebda.

Niedersachsen: Rahmenrichtlinien für das Gymnasium, Biologie, Gymnasiale Oberstufe, Erl. d. KM. v. 18.4.1984, GUELT.: 152/263, Hannover, 1984, S. 19 und 26-27

Nordrhein-Westfalen: Richtlinien für das Gymnasium in Nordrhein-Westfalen. Biologie. Gymnasiale Oberstufe, Richtlinien Biologie, RDRL. d. KM v. 16.6.1981, GABL NW, S. 199, Köln 1982, S. 13

Schleswig-Holstein: Lehrplan Gymnasium, Biologie, Oberstufe, Kiel 1986, S. 34-39

27) Bayern: Curricularer Lehrplan für das Fach Biologie in der Kollegstufe des Gymnasiums. KMBL I, So.-Nr. 17/1979, München, 1979

Weiterhin soll das "Bewußtsein der Bedeutung und Folgen von Eingriffen des Menschen in Ökosysteme" (Lernziel 5.3 bzw. 2.4) anhand der Inhalte "Belastbarkeit" und "Selbstreinigungskraft" der Gewässer, Veränderung der Nährsalzzufuhr durch Industrie, Haushalt und Landwirtschaft (Eutrophierung, Sauerstoffschwund) sowie Abwasserreinigung aufgebaut werden.

Es kann als Ergebnis der Durchsicht festgestellt werden, daß sich die für das Computermodell *LAGO* vorgesehene Zielsetzung mit den Intentionen in den verschiedenen Lehrplänen deckt.

Was die Hardware anbetrifft, so soll das Computermodell *LAGO* zunächst für die Rechnerklasse der IBM*-kompatiblen Computer (vgl. Anmerkung in Fußnote 14, S. 23) entwickelt werden. Für den Einsatz des Modells im Unterricht ist also ein Computer aus dieser Klasse, das Betriebssystem MS-DOS ab Version 2.x, ein Diskettenlaufwerk (5¼ Zoll, 360 KByte) und ein Arbeitsspeicher von mind. 128 KByte notwendig. Zur Mindestausstattung gehören außerdem ein grafikfähiger Monitor und der entsprechende Grafikadapter. Mit dieser Konfiguration wird zunächst ein großer Teil der neueren, in den Schulen angeschafften Rechner erfaßt. Als Folgearbeit ist vorgesehen, das Programm dann an weitere leistungsfähige, häufig vorhandene Computer anzupassen.

Die Entwicklung des Computermodells *LAGO* erfolgt, wie bereits angemerkt, anhand des in Kapitel 4 vorgestellten Verfahrens. Im weiteren Verlauf der Arbeit werden die hierbei nötigen Entwicklungsschritte ausführlich dargestellt und die entstehenden Schwierigkeiten diskutiert. Im nächsten Abschnitt ist es daher zunächst erforderlich, als Teil der Sachanalyse, das Ökosystem "See" aus biologischer und systemtheoretischer Sicht zu charakterisieren. Um Wiederholungen zu vermeiden, werden weitere der vielen, für die Modellierung notwendigen fachlichen Überlegungen dort genannt, wo sie für Entscheidungen gebraucht werden, z.B. bei der Erstellung des Wortmodells und des formalen Modells.

5.2. Charakterisierung des Ökosystems "See"

Charakterisierung des Ökosystems "See" aus biologischer Sicht

Seen unterscheiden sich als aquatische Ökosysteme durch verschiedene Eigenschaften und Funktionen deutlich von den sie umgebenden terrestrischen. Einige für die Modellentwicklung wesentliche Unterschiede seien im folgenden zusammengestellt:

Im See wird die Primärproduktion - abgesehen von der unmittelbaren Uferzone (Litoral) oder von Seen mit sehr geringer Wassertiefe - von mikroskopisch kleinen Algen mit extrem geringer individueller Biomasse geleistet. Die Lebensdauer des Phytoplanktons bewegt sich im Zeitraum von Stunden bis wenigen Tagen. Gemäß der Regel, daß die Umsatzrate bei Pflanzen wie Tieren mit zunehmender Körpergröße sinkt und umgekehrt, ist die Primärproduktion durch sehr hohe Umsatzraten gekennzeichnet. "Innerhalb der an Land für Produktionsmessungen möglichen Zeitabschnitte (z.B. Frühjahr bis Herbst) können im Pelagial eines Sees von einigen Hundert bis zu vielen Tausend Generationen aufeinander gefolgt sein. Im gleichen Zeitraum können aber auch einige Arten verschwunden und andere hinzugekommen sein. Schließlich gilt für aquatische Biotope, insbesondere für den Lebensraum des Pelagials, daß die Verluste durch Tierfraß in der Regel um ein Vielfaches höher sind als an Land. So ist es verständlich, daß zwischen einem etwaigen Biomassezuwachs innerhalb einer Vegetationsperiode (z.B. Frühjahr bis Herbst) und der in diesem Zeitraum erfolgten Produktion keine unmittelbare Beziehung besteht..." (SIEBECK 1982, S. 119)

Während terrestrische Ökosysteme durch "festgewachsene Primärproduktion" (Gräser, Kräuter, Sträucher und Bäume) gekennzeichnet sind, ist der größte Teil der pflanzliche Biomasse im See - wiederum mit Ausnahme des Litorals - beweglich. Es herrscht eine vertikale Verteilung der Primärproduktion vor, die im Prinzip von den Faktoren Größe des Energieangebots, Charakteristik des Strahlungsabfalls mit wachsender Tiefe und Größe der Maximalproduktion im Optimum des Strahlungsangebots abhängt (vgl. SIEBECK 1982, S. 120). Die Struktur des Seenbeckens beeinflusst in starkem Maße die Produktionsverhältnisse.

Auch die Nährstoffe sind mobil. Sie werden von den Wasserströmungen, durch Diffusionsprozesse und durch Temperaturumschichtungen im Wasserkörper verteilt. Eingebraachte Stoffe können sich daher im See-Ökosystem ungleich stärker ausbreiten und verteilen als unter terrestrischen Verhältnissen.

Während sich bei den Gegebenheiten an Land die eingebrachten Stoffe im Boden viel weniger rasch ausbreiten können als im Wasser, bringt es die Abgeschlossenheit des Seebeckens mit sich, daß, je nach morphometrischen Gegebenheiten, der Wasserkörper mehr oder weniger stark zur Anreicherung von Nährstoffen (Eutrophierung) neigt. Nur wenn Zufluß und Abfluß groß sind im Vergleich zum Seevolumen, können sich Nährstoffzufuhr und -austrag die Waage halten.

Aus der oben erwähnten hohen Umsatzrate folgt ein rapider Energiedurchzug, da dieser der Umsatzrate direkt proportional ist. Der Energiedurchzug im Ökosystem "See" wird häufig weitgehend von der eingestrahlten Lichtenergie bestimmt. Im Gegensatz zum Fließgewässer tritt die kontinuierliche Energie der Strömung stark zurück; dafür gewinnen aber Temperaturschwankungen und Wind an Bedeutung.

Charakterisierung des Ökosystems "See" aus systemtheoretischer Sicht

Das Ökosystem "See" stellt, wie alle Systeme, "einen nach bestimmten Gesichtspunkten abgegrenzten Bereich der objektiven Realität dar, der aus kleineren Einheiten, den *Elementen* (Kompartimente) besteht, die untereinander durch *Relationen* (Kopplungen) verbunden sind. Bei den Relationen sind zwei wichtige Kopplungsrelationen zu unterscheiden, die *direkte Kopplung* (feed forward) und die *Rückkopplung* (feed-back). Der nicht zum System gehörende Teil heißt *Umwelt*. System und Umwelt werden durch eine endliche Hülle, den *Systemrand*, voneinander getrennt. Die Abgrenzung des Systems ist relativ; dieselben Elemente können, je nach Standpunkt, zum System oder zur Umwelt gehören. Meist wird aber der Systemrand so gewählt, daß sich die Mehrzahl der Relationen, besonders die Rückkopplungen, innerhalb des Systems abspielt und nur einige zur Umwelt führen." (sinngemäß übersetzt aus STRASKRABA/GNAUCK 1985, S. 14) Beim See bilden die miteinander vergesellschafteten

Populationen einschließlich der "zugehörigen" Umweltfaktoren (z.B. Licht, Sauerstoff) ein Ökosystem, während das Festland die Umwelt des Systems darstellt. Der Systemrand wird durch die Uferlinie repräsentiert.

Das Ökosystem "See" gehört, wie alle Ökosysteme, zu den sogenannten "Großen Systemen"; es besitzt von daher eine Reihe typischer Eigenschaften, von denen im folgenden einige wesentliche genannt werden (vgl. zu den folgenden Punkten STRASKRABA/ GNAUCK 1985, S.14, WEDEKIND 1981, S. 34):

1. Das System steht mit der Umgebung im Materie- und Energieaustausch. Ein solches System wird als *offen* bezeichnet.
2. Das System enthält eine große Zahl qualitativ verschiedener Elemente. Menge und Art der Relationen der Systemelemente untereinander sowie zwischen dem System und seiner Umwelt sind sehr groß. Ein solches System wird als *komplex* bezeichnet. Die Strukturen sind bestimmt durch physikalische Bedingungen (z.B. räumliche Verteilung, Licht- und Energiebedingungen), chemische Bedingungen, (z.B. chemische Umsetzungen in den Stoffkreisläufen), biologische Bedingungen (z.B. trophische Ebenen, Diversität) und zeitliche Struktur (z.B. biologische Sukzession, Systemevolution). Die Funktionen des Systems sind verknüpft mit den Kreisläufen von Biomasse und Energie, mit den Wechselbeziehungen innerhalb der Systemelemente wie auch den Wechselbeziehungen zwischen System und Umwelt. Spezielle ökologische Funktionen (z.B. Konkurrenz, Räuber-Beute-Beziehungen, Synchronisation) sind zwischen den miteinander vergesellschafteten Populationen und auf den verschiedenen trophischen Ebenen vorhanden.
3. Das System besitzt Eigenschaften, die erst im Ergebnis der Wechselwirkungen der Systemelemente entstehen. Dieses Merkmal wird als *Integrität* bezeichnet.
4. Struktur und Zustand des Systems sind räumlich und zeitlich veränderbar. Das Ökosystem "See" gehört daher zu den *dynamischen* Systemen.
5. Das System besitzt Puffer- und Speichervermögen. Bei Einwirkung von Störungen verläuft der Übergang aus einem Zustand in einen anderen nicht sprunghaft, sondern er wird durch einen sich entwickelnden (Übergangs-)

Prozeß charakterisiert. Wird das System aus der Gleichgewichtslage gebracht, kehrt es wieder in diesen Gleichgewichtszustand zurück. Diese Eigenschaft wird als *stabil* bezeichnet. Im System herrscht ein *Fließgleichgewicht*, in dem durch ständigen Materie- und Energiefluß Struktur und Ordnung aufrechterhalten werden.

6. Das System ist nicht deterministisch festgelegt, sondern zählt zu den *stochastischen* Systemen. Einer Eingabe (input) kann daher nicht eindeutig eine bestimmte Ausgabe (output) zugeordnet werden; die Ausgabe wird zusätzlich von systeminternen Zufallsfaktoren bestimmt.

Die systemtheoretische Beschreibung zeigt, daß das Ökosystem "See" also durch die Merkmale *offen*, *komplex*, *dynamisch*, *stabil* und *stochastisch* gekennzeichnet ist. Bei der Modellentwicklung sollten diese Eigenschaften erhalten bleiben. Dies stellt aber sehr hohe Anforderungen an die Modellierung. Besondere Schwierigkeiten bereiten die Merkmale "offen", "komplex" und "stochastisch"; bei der Konzeption von Ökosystem-Modellen müssen hier meist die stärksten Vereinfachungen vorgenommen werden. Die Offenheit kann z.B. dadurch etwas reduziert werden, daß ein See mit geringem Wasserdurchfluß angenommen wird.

5.3. Erstellung des Wortmodells

Wie in Kapitel 4 (S. 27) ausgeführt, kommen die im Wortmodell enthaltenen Vereinfachungen durch didaktische Reduktion zustande. Für das folgende Wortmodell ist, neben diesem didaktischen Aspekt, noch ein weiterer Grund für die Vereinfachung maßgeblich. Wie Vorarbeiten, die bei Punkt 5.4.1. (S. 50) noch näher ausgeführt werden, gezeigt haben, sind der Modellierung durch die Arbeitsgeschwindigkeit des Computers deutliche Grenzen gesetzt, obwohl die bereits genannten moderneren Arbeitsplatzcomputer (vgl. Fußnote 14, S. 23) verwendet wurden. Die Beschränkungen betreffen in erster Linie das formale Modell; sie sind aber bei der folgenden Beschreibung des Wortmodells bereits mitbedacht.

Struktur des Modellsees

Von den in Frage kommenden *Zonen* eines Sees soll nur das Pelagial modelliert werden. Diese Vereinfachung ist ohne weiteres vertretbar, da das Pelagial für das Zusammenwirken physikalischer, chemischer und biologischer Faktoren im Ökosystem besonders charakteristisch ist, sich die didaktischen Zielsetzungen also hieran exemplarisch verwirklichen lassen. Würde das Litoral in das Modell einbezogen werden, müßten zusätzliche, für diese Zone typische Systemelemente, wie z.B. die litoralen Makrophyten und das Periphyton, Berücksichtigung finden. Ein zur Benthalthalregion gehörender Teil, das Profundal, soll allerdings indirekt in die Modellierung einbezogen werden: Es ist beabsichtigt, über die sich im Sediment ablagernde Biomasse die Sauerstoffzehrung zu ermitteln. Die Begründung hierfür erfolgt in Abschnitt 5.4.9. (S. 92).

Das Modellökosystem soll aus folgenden *Systemkomponenten* zusammengesetzt sein: Stellvertretend für die Lebensgemeinschaft des Pelagials stehen Phytoplankton und Zooplankton. Die Rolle der Nährstoffe im Stoffhaushalt des Sees wird exemplarisch am reaktiven, gelösten (Ortho-)Phosphat aufgezeigt. An der räumlich und zeitlich unterschiedlichen Verfügbarkeit des Sauerstoffs für die Organismen im Pelagial soll die Auswirkung unterschiedlicher Produktivität eines Sees deutlich gemacht werden. Licht und Wassertemperatur sollen den Einfluß abiotischer Faktoren zeigen. Eine wesentliche Vereinfachung gegenüber der Realität besteht darin, daß keine Modellierung verschiedener Populationen des Phyto- und Zooplanktons sowie für die verschiedenen Altersklassen innerhalb einer bestimmten Zooplanktonpopulation erfolgt. Das Modell kann daher Sukzessionen, die durch Konkurrenz und Selektion hervorgerufen werden, nicht beschreiben. Auch sind die für den Abbau der organischen Stoffe verantwortlichen Bakterien nicht direkt enthalten. Der didaktische Wert des Computermodells bleibt trotzdem voll bestehen, da sich mit Hilfe der genannten Systemkomponenten die erwünschten grundlegenden Einsichten in die wesentlichen Zusammenhänge im Ökosystem vermitteln lassen.

Von den in Frage kommenden *Seentypen* soll der für die gemäßigte Klimazone typische dimiktische See modelliert werden. Die für diesen See charakteristische Abfolge von Zirkulation und Stagnation ist besonders gut geeignet, die Auswirkungen physikalischer Faktoren auf das Ökosystem zu zeigen. Die Mo-

dellierung der thermischen Schichtung soll anhand eines idealisierten Jahresverlaufs der Tiefenverteilung der Wassertemperatur erfolgen.

Vorgänge und Zusammenhänge

Bei der Modellierung der Vorgänge und Zusammenhänge ist vor allem darauf zu achten, daß die einzelnen Prozesse nicht voneinander isoliert betrachtet und nicht nur linear angeordnet werden. In der folgenden Konzeption sind daher viele Rückkopplungen enthalten, damit die wechselseitigen Abhängigkeiten der verschiedenen Faktoren verdeutlicht werden können.

Im Frühjahr vermehren sich die Algen infolge steigender Wassertemperatur und Lichtintensität sehr stark. Es wird angenommen, daß die Konzentration des anorganischen, gelösten Orthophosphats durch die erfolgte Frühjahrszirkulation ausreichend hoch ist, so daß nur die Temperatur als limitierender Faktor wirkt. Voraussetzung ist natürlich ein ausreichendes Angebot an allen anderen essentiellen Nährstoffen. Das filtrierende Zooplankton ist noch sehr gering entwickelt; die Verluste von Algenbiomasse durch Zooplanktonfraß fallen noch kaum ins Gewicht.

Mit Beginn der Sommermonate bildet sich allmählich durch die weitere Zunahme der Wassertemperatur an der Oberfläche des Sees eine thermische Schichtung des Wasserkörpers aus. Bei seiner Photosynthesetätigkeit inkorporiert das Phytoplankton einen erheblichen Anteil des anorganischen Orthophosphats, da infolge der schichtungsbedingten Trennung von Oberflächen- und Tiefenwasser der Phosphatnachschieb aus dem Hypolimnion weitgehend verhindert wird. Damit wirkt die jetzt stark verringerte Orthophosphatkonzentration limitierend auf das Phytoplanktonwachstum. Algensedimentation und erhöhter Zooplanktonfraß stellen wesentliche Verlustgrößen dar, so daß nun die Gesamtbilanz des Phytoplanktonwachstums negativ wird. Abbauprozesse am Grund des Sees führen zu einem, je nach trophischem Zustand, räumlich und zeitlich unterschiedlichen Sauerstoffdefizit in der tropholytischen Zone.

Im weiteren Verlauf wird Phosphat von den Algen und vom Zooplankton ausgeschieden (kurzgeschlossener Phosphat-Kreislauf), die Orthophosphatkonzentration steigt an, die Wachstumsbilanz der Algen gleicht sich aus, bei günsti-

gen Bedingungen kommt es zu einem "Herbstgipfel". Später führen fallende Temperaturen und abnehmende Strahlungsintensität zu einem Absinken der Nettoproduktion. Die Biomassen der Algen und des Zooplanktons nehmen ab, das Phosphat wird wieder remineralisiert. Infolge der Abkühlungen der oberen Wasserschichten hat sich die sommerliche Schichtung abgebaut und der gesamte Wasserkörper nimmt allmählich eine Temperatur von 4 °C - 5 °C an. Unter dem Einfluß der Herbststürme kommt es zur erneuten Vollzirkulation des Sees. Damit gleichen sich die ehemals unterschiedlichen Konzentrationsverhältnisse des Orthophosphats wieder aus, Sauerstoff gelangt bis in die untersten Schichten des Wasserkörpers.

Im Winter kommt es dann zu einer inversen thermischen Schichtung des Sees. Dieser Schichtung fehlt das eigentliche Epilimnion und die vertikale Temperaturschichtung beginnt sozusagen mit einer inversen Sprungschicht. Der Sauerstoff wird in der Regel nicht aufgebraucht, da der bakterielle Abbau bei niederen Temperaturen nicht sehr hoch ist und Wasser bei geringeren Temperaturen mehr Sauerstoff löst.

Zusätzlich zu diesen, den oligotrophen oder leicht mesotrophen See kennzeichnenden Vorgängen soll das Computermodell *LAGO* die typischen Folgen anthropogen bedingter Eutrophierung aufzeigen.

Hierzu wird der Modellsee durch einen Phosphatimport über den Zufluß des Sees belastet. Die Größe dieses Phosphatimports soll der Anwender selbst festlegen können, indem er in den Computer bestimmte Werte eingibt, aus denen sich dann die Phosphatbelastung des Sees errechnet. Folgende Werte sind vorgesehen: die Einwohnerzahl einer fiktiven, am Seeufer angesiedelten Gemeinde, die Fläche der direkt an den See grenzenden, gedüngten Acker und Wiesen, der Prozentsatz der Einwohner, die phosphathaltige Waschmittel verwenden und die Art der Behandlung des Abwassers. Bei Letzterer stehen als Alternativen die direkte Einleitung in den See und die Abwasserreinigung mit Hilfe einer Kläranlage zur Disposition (siehe Anlage 5, S. 158).

Der erhöhte Phosphatimport hat zur Folge, daß keine das Phytoplanktonwachstum limitierende Phosphatkonzentrationen auftreten. In der Tiefe des Strahlungsoptimums wird die Produktion und damit die Individuendichte des Phytoplanktons, eine identische Artenzusammensetzung vorausgesetzt, besonders

stark zunehmen und das Strahlungsangebot in den darunter liegenden Wasserschichten vermindern ("Selbstbeschattung" der Algen). Dort wird somit die weitere Produktion abnehmen. Da die Produktion mit steigendem Nährstoffangebot aber nicht nur innerhalb des Strahlungsoptimums, sondern auch in den darüber liegenden Schichten zunimmt, wird die Lage des Produktionsmaximums immer weiter zur Wasseroberfläche wandern. Die trophogene Zone wird schmaler.

Mit der Biomasse der Algen steigt die Biomasse des Zooplanktons. Es ergibt sich eine Zunahme der Masse des sedimentierenden Planktons, was zur Folge hat, daß sich der Sauerstoffverbrauch durch die aeroben, diese Biomasse mineralisierenden Bakterien erhöht. Damit entsteht, vor allem in den untersten Wasserschichten, ein Sauerstoffmangel, der schließlich sogar zu anaeroben Verhältnissen führen kann. Die Abbauprozesse (Fäulnis, Gärung) übernehmen nun anaerobe Bakterien, wobei giftige Gase, wie z.B. Schwefelwasserstoff, produziert werden.

Bei einer länger andauernden Phase anaerober Verhältnisse über dem Wassergrund ist außerdem die Gefahr der Phosphatfreisetzung aus dem Sediment ("interne Düngung") gegeben. Folge ist eine Selbstverstärkung der Eutrophierung. Im Extremfall wird der Sauerstoff bis in die oberen Schichten des Wasserkörpers völlig verbraucht. Wenn bei einer solchen Situation die Herbstvollzirkulation einsetzt, kann es geschehen, daß in der trophogenen Zone die Konzentration des Sauerstoffs sehr gering, die der giftigen Gase aber sehr hoch wird, so daß die Lebensbedingungen für die höheren Organismen nicht mehr gegeben sind: der See ist "umgekippt". Die Simulation soll dann unter Hinweis auf diese Verhältnisse abgebrochen werden.

5.4. Erstellung des formalen Modells

5.4.1. Vorüberlegungen

Auswahl des Modelltyps

LEVINS (1966) unterteilt die Modelle nach drei Haupteigenschaften: Präzision, Realität und Generalität. Demzufolge lassen sich drei Modellgruppen unterscheiden ²⁸⁾:

1. Modelle, welche die *Präzision* gegenüber der Realität und der Generalität hervorheben. Diese Modelle können kurz als präzise Modelle bezeichnet werden. Hier sind die empirischen statistischen Modelle und solche Modelle einzuordnen, die mittels stochastischer Methoden gebildet wurden, wie sie z.B. GNAUCK (1974) für die Talsperrenproblematik ausgearbeitet hat. Modelle dieser Modellgruppe finden z.B. in der Wasserwirtschaft (Waterquality Management) Anwendung.

2. Modelle, die die *Realität* gegenüber der Präzision und der Generalität betonen. Diese Modelle werden als reale Modelle bezeichnet. Hierzu gehören besonders diejenigen, welche die sich im Ökosystem abspielenden Prozesse in quantitativer Form ausdrücken. Es handelt sich um analytische Modelle, die für Talsperrenökosysteme von DI TORO et al. (1971) begründet wurden. Modelle dieser Modellgruppe eignen sich besonders für die wissenschaftliche Analyse der Prozesse in Ökosystemen und ihre Quantifizierung.

3. Modelle, welche die *Generalität* gegenüber Präzision und Realität betonen (a priori - Modelle). Zu diesem Typ gehören Modelle, die das Verhalten eines Ökosystems auf einer allgemeinen mathematischen Ebene auszudrücken versuchen (für aquatische Ökosysteme siehe HARTMANN und BOES, 1973). Die Generalität eines Ökosystemmodells kann aber auch vom Gesichtspunkt der bedeckten Breite der Bedingungen (z.B. der geographischen Bedingungen) aus betrachtet werden.

²⁸⁾ Zur folgenden Einteilung muß angemerkt werden, daß sie nicht absolut trennscharf ist. Die Grenzen der einzelnen Modellgruppen gehen mehr oder weniger kontinuierlich ineinander über.

Optimal wäre nun ein Modell, das alle oben genannten Aspekte kombiniert. Dies ist aber nur in gewissem Maße möglich. Wegen der überaus hohen Komplexität der Beziehungen zwischen Pflanzen und Tieren und der Schwierigkeit, diese Beziehungen zu erkennen und zu messen, kamen einige mathematisch arbeitende Ökologen zu dem Schluß, daß Modelle nicht gleichzeitig präzise, realistisch *und* allgemeingültig sein können (LEVINS 1966). Je nach beabsichtigtem Modellierungszweck muß daher einer der drei genannten Aspekte in den Vordergrund gestellt werden.

Für die weitere Modellierung muß daher entschieden werden, welcher der drei Modelltypen für die vorgesehenen didaktischen Zielsetzungen am geeignetsten ist. Im Vordergrund steht zweifellos das Prinzip der Generalität (Allgemeingültigkeit). Dies ergibt sich aus der im Wortmodell präzisierten Zielsetzung: Das Computermode'll soll einige ausgewählte Zusammenhänge in einem See, der den unseren klimatischen Bedingungen entsprechenden Seentyp repräsentiert, veranschaulichen. Eine reale Entsprechung zu einem bestimmten See ist daher nicht vorrangige Absicht. Aufgrund der aus didaktischen Gründen erforderlichen Vereinfachungen ist es natürlich auch nicht sinnvoll, hohe Anforderungen an die Präzision des Modells zu stellen.

Auswahl der Methode

In Kapitel 4 (S. 27 ff.) wurden die beiden für die Erstellung eines formalen Modells in Frage kommenden Methoden vorgestellt. Für das Computermode'll *LAGO* soll die *analytische Modellierung* angewandt werden, da sie sich für die Simulation von stabilen Systemen als Standardmethode etabliert hat. Es gibt daher zahlreiche, für Zwecke der Systemforschung oder der Wasserwirtschaft (water quality management) entwickelte Simulationsmodelle zu aquatischen Ökosystemen, an denen sich die Arbeit orientieren kann. Ein weiterer Grund für die Entscheidung zugunsten der analytischen Modellierung liegt in der im Wortmodell festgelegten Absicht, daß das Computermode'll auch typische Zusammenhänge der anthropogen bedingten Eutrophierung eines Sees veranschaulichen soll. Nach STRASKRABA/GNAUCK kann dieser Prozeß aber nur mit den Mitteln der analytischen Modellierung quantitativ erfaßt werden: "Analytical models are the only instruments, at present, for an adequate assessment of eutrophication ..." (STRASKRABA/GNAUCK 1985, S. 225).

Die Basis des Computermodells bildet daher ein mathematisches Modell. Es setzt sich im allgemeinen aus drei Gruppen von Systemvariablen zusammen: Zustandsgrößen, Störgrößen und Umgebungsgrößen. Nach STRASKRABA/GNAUCK (1985, S. 49/50) sind die Größen wie folgt zu charakterisieren:

Die Zustandsgrößen (Bestandsgrößen, Speichergrößen, state variables) stellen die Kompartimente des Modells dar und beinhalten die Speichereigenschaften und Rückkopplungen des Systems. Jede Zustandsgröße ändert sich auf Grund des Gesetzes der Erhaltung von Masse und Energie, ihre zeitlichen Änderungen werden mittels Differentialgleichungen (Bilanzgleichungen) beschrieben. Die in einer solchen Bilanzgleichung auftretenden Geschwindigkeiten der Zuflüsse, Zunahmen, Verluste und Abflüsse (rate constants) sind spezifische, auf die jeweilige Masseneinheit bezogene Werte. Im Modell *LAGO* sind drei Zustandsgrößen, die Phytoplanktonbiomasse, die Zooplanktonbiomasse und die Orthophosphatkonzentration, jeweils bezogen auf die durchmischte Schicht, sowie zahlreiche Veränderungsraten und Hilfsvariablen enthalten.

Die *Störgrößen* (driving variables, forcing functions), oft auch als Eingangsgrößen (inputs) bezeichnet, stören von außen die Änderungen der Zustandsgrößen im System, sie sind aber von ihnen unabhängig. Das Modell *LAGO* berücksichtigt folgende typische Störgrößen aquatischer Ökosysteme: Wassertemperatur, Lichtstrahlung, Tageslänge und Phosphatkonzentration im Zufluß. Ihre mathematische Beschreibung wird durch eine Funktion $y = f(t)$ ausgedrückt, da eine Approximation durch konstante Mittelwerte der Zeitabhängigkeit dieser Größen nicht entspricht. Im einfachsten Fall liegt $y = f(t)$ als Wertetabelle vor. Dies ist z.B. für die Modellierung der thermischen Schichtung vorgesehen.

Die *Umgebungsgrößen* (site constants) charakterisieren die physikalischen Grenzen, in welchen sich die zu modellierenden Prozesse abspielen. Für limnische Ökosysteme sind dies die morphometrischen und hydrologischen Eigenschaften sowie die innere Struktur des betreffenden Gewässers. Erstere umfassen im Modell *LAGO* die Form des Seebeckens, die Oberfläche des Sees, das Volumen des Wasserkörpers, die maximale Tiefe und die Durchmischungstiefe. Die hydrologische Umgebungsgröße "Wassererneuerungszeit", die z.B. für physikalische, chemische und biologische Prozesse im See eine große Rolle spielt, ist im Modell konstant (vgl. Punkt 5.4.2., S. 59). Die einfach gehaltene innere

Struktur besteht in einer Zerlegung des Wasserkörpers in zwei homogene Schichten (durchmischte Schicht und Hypolimnion, vgl. Anlage 1, S. 133).

Das Kernstück des mathematischen Modells soll also ein System von miteinander gekoppelten Differentialgleichungen bilden. Ergänzt wird dieses Grundmodell durch Fallentscheidungen und Wertetabellen. In Anlage 1, S. 133 sind die genannten Größen und ihre Zusammenhänge in einem vereinfachten Schema dargestellt.

Für die numerische Berechnung der mit einem Digitalrechner auf analytischem Weg nicht lösbaren Differentialgleichungen soll das Verfahren von RUNGE-KUTTA (vgl. BRONSTEIN/SEMENDJAJEW 1980) verwendet werden. Es beruht auf der Approximation der analytischen Lösung in der Nähe jedes Knotenpunktes durch Polynome verschiedener Ordnung. Vorteile des Verfahrens liegen in der relativ hohen Genauigkeit und in der Möglichkeit, die Schrittweite während des Verfahrens zu ändern, wenn dies wegen des Verlaufs der Lösungskurve zweckmäßig erscheint. Bei der numerischen Lösung von Differentialgleichungen entstehen ein *Approximationsfehler* sowie ein durch die Arbeit des Rechners mit einer beschränkten Zahl von Dezimalstellen bedingter *Rundungsfehler*. Da der Approximationsfehler u.a. von der Schrittweite h abhängt, erhalten wir durch Verkleinerung von h eine verbesserte Lösung. Die Rechenzeit steigt dann aber beträchtlich an. Einen Ausweg aus diesem Dilemma bieten R-K-Modifikationen, bei denen die Schrittweite automatisch verringert wird, wenn die Näherungslösung einen bestimmten Fehler überschreitet.

Parametrisierung

Die Bestimmung der im mathematischen Modell erforderlichen Parameter soll anhand zahlreicher Simulationsläufe erfolgen, bei denen festgestellt wird, mit welcher Sensitivität das Modell auf die Veränderungen einzelner Parameter reagiert. Dieses Verfahren kann durch den Computer unterstützt werden: Nachdem das mathematische Modell in ein Computerprogramm umgewandelt worden ist, werden in dieses Programm Testprozeduren eingebaut, die einen zu bestimmenden Parameter vom gewünschten Anfangswert bis zum gewünschten Endwert in beliebiger Schrittweite verändern und die berechneten Werte der abhängigen Variablen in einer Tabelle festhalten. Bei Verwendung mehrdimen-

sionaler Tabellen lassen sich gleichzeitig mehrere Parameter systematisch verändern. Die Daten werden anschließend analysiert und der günstigste Parameter ermittelt. Für diese Auswertungsarbeit haben sich professionelle Tabellenkalkulationsprogramme, in deren Leistungsumfang statistische Funktionen enthalten sind oder die die Eingabe mathematischer Formeln für die Datenanalyse erlauben, gut bewährt.

Ergebnisse von Vorarbeiten

Als Vorarbeiten zur Modellentwicklung habe ich einige mathematische Modelle, die die Eutrophierung von Seen für Zwecke der Ökosystemforschung oder der Wasserwirtschaft quantitativ erfassen, in lauffähige Computerprogramme umgesetzt. Im einzelnen handelt es sich um die Modelle:

- Eutrophierungsmodelle AQUAMOD 1, AQUAMOD 2, AQUAMOD 3 (STRASKRABA/GNAUCK 1985)
- A dynamic lake model for trophic state prediction (IMBODEN/GÄCHTER 1978)
- Modelling of grangent reservoir ecosystems (GARÇON 1981)
- Experimente mit einem dynamischen ökologischen Modell der Freiwasserregion von Talsperren und Seen (BENNDORF/RECKNAGEL 1979)
- Three layer model of an aquatic ecosystem (DVORAKOVA/KOZERSKI 1980)
- Ein mathematisches Ökosystemmodell für eutrophe Flachgewässer (SCHELLENBERGER u.a. 1983)
- A diskrete model for the study of a lake (GENTIL 1979).

Die aufgelisteten Modelle sind von unterschiedlicher hydrodynamischer und biologischer Komplexität und gehören zur Klasse der analytischen, rein deterministischen Modelle mit einer Dynamik erster Ordnung. Sie beschreiben das reale Ökosystem, mit Ausnahme des letztgenannten diskreten Modells von GENTIL, mittels einfacher, gekoppelter, nichtlinearer Differentialgleichungen mit fester Struktur. Sie besitzen keine Anpassungsmechanismen, berücksichtigen aber Rückkopplungsmechanismen.

Bei der genannten Dynamik erster Ordnung, auch Kinematik genannt, laufen die Stoffkreisläufe ohne Änderung der Geschwindigkeit ab. Dies wird als "Dynamik des Systemzustandes" bezeichnet. In einem mathematischen Modell wird diese Tatsache durch eine feste Struktur S und einen festen Parametervektor p beschrieben:

$$dS/dt = f(S, p, S_0, t)$$

S : feste Zustandsvariable

S_0 : Anfangswert von S

p : Vektor des festen Geschwindigkeitsparameters

t : Zeit

Ein Modell mit der Dynamik erster Ordnung entspricht der Modellklasse mit direkter oder rückgekoppelter Steuerung.

Ziel der Vorarbeiten war es, Erfahrungen zu gewinnen, welche der in den verschiedenen Submodellen verwendeten mathematischen Beziehungen sich für die erwünschten allgemeinen Aussagen über das Ökosystem "See" eignen könnten, mit welcher Laufzeit bei einem bestimmten Modellumfang zu rechnen ist und welche programmiertechnischen Besonderheiten, etwa zur Auswahl der Programmiersprache, zu erwarten sind.

Bei den Vorarbeiten ergab sich ein für die Modellentwicklung gravierendes Problem. Es betrifft die Laufzeit des Simulationsprogramms. Setzt man für eine Unterrichtsstunde, in der mit dem ComputermodeLL gearbeitet wird, großzügig 90 Minuten an, so sollten in diesem Zeitraum wenigstens zehn Modelljahre simuliert werden können, damit sich die Auswirkungen unterschiedlicher klimatischer Verhältnisse und eine unterschiedliche Phosphatbelastung sinnvoll erproben lassen. Außerdem ist ausreichend Zeit für Auswertung und Diskussion notwendig (ca. 30 Minuten). Daraus folgt aber, daß für ein Modelljahr maximal 6 Minuten (!) zur Verfügung stehen. Für die vorgesehene numerische Berechnung eines komplexeren Differentialgleichungssystems ist das aber extrem wenig. Bei der Erprobung der oben genannten Simulationsmodelle waren zwischen 30 Minuten und 3 Stunden für ein einziges Modelljahr erforderlich.

Dieses Laufzeitproblem ist für Simulationsprogramme, die auf den genannten Arbeitsplatzrechnern laufen und für den Unterricht entwickelt werden sollen,

typisch. Als Lösung sind verschiedene Wege denkbar. Eine Reduktion des mathematischen Modells auf ein Niveau, bei dem solche Laufzeitschwierigkeiten nicht mehr auftreten, ist ausgeschlossen, weil ein solches Modell sehr einfach sein müßte und sich damit die didaktische Zielsetzung, einen Einblick in die komplexen Zusammenhänge in einem Ökosystem zu ermöglichen, nicht mehr verwirklichen ließe. Ein anderer Weg besteht darin, Abstriche bei der Rechengenauigkeit vorzunehmen. Es kann z.B. die Zahl der gültigen Dezimalstellen reduziert, im Extremfall sogar nur mit ganzen Zahlen gerechnet werden oder man wählt bei der numerischen Berechnung analytisch nicht lösbarer Differentialgleichungen eine große Schrittweite. Damit läßt sich, wie die Vorarbeiten gezeigt haben, einiges an Rechenzeit einsparen. Diese Lösung enthält aber erhebliche Gefahren. Wenn man z.B. bei der Simulation stark ansteigender oder abfallender Wachstumsraten zu starke Einbußen bei der Rechengenauigkeit in Kauf nimmt, kann ein einigermaßen realistisches, Anspruch auf Allgemeingültigkeit erhebendes Modellverhalten nicht mehr erreicht werden. Bei einem diskreten Modell treten prinzipiell die gleichen Schwierigkeiten auf: Auch hier führt die Vergrößerung der Zeitschritte zwecks Laufzeitverminderung zu einer nicht mehr akzeptablen Modellgüte. Das Laufzeitproblem läßt sich natürlich auch durch eine leistungsfähigere Hardware, z.B. durch die Verwendung eines zusätzlichen mathematischen Prozessors oder schnellerer Arbeitsplatzrechner entschärfen ²⁹⁾. Diese Möglichkeit soll aber ausgeklammert werden, da die Schulen damit nicht ausgestattet sind.

Aufgrund dieser Überlegungen bietet sich für die Entwicklung des Computermodells *LAGO* folgender Weg an: Das mathematische Modell wird auf wenige, unbedingt erforderliche Zustandsgrößen beschränkt, damit die Zahl der Differentialgleichungen gering bleibt. Andererseits ist eine hohe Zahl von Rückkopplungen vorgesehen, um die vielfältigen Wechselwirkungen im Ökosystem "See" aufzeigen zu können. Diese Lösung orientiert sich an einer Studie von JÖRGENSEN et al. (1978), wonach die Generalität und Realität eines Modells nicht unbedingt von seiner Detailliertheit, z.B. der Zahl der Zustandsvariablen, abhängt. Eher ist es die Zahl der im Modell enthaltenen Rückkopplungen, die die Modellgüte verbessert. Für die numerische Berechnung der analytisch nicht lösbaren Differentialgleichungen soll eine flexible Schrittweitensteuerung verwendet werden: Wenn z.B. eine der im Modell vorhandenen Veränderungsraten

29) Bei den Arbeitsplatzrechnern gibt es z.B. die sog. AT (Advanced Technology) - Klasse mit dem Mikroprozessor 80286. Auch Geräte mit einer noch höheren Arbeitsgeschwindigkeit (Mikroprozessor 80386, getaktet mit 16 oder sogar 20 MHz) werden schon ausgeliefert.

im Begriff ist, deutlich anzusteigen oder abzufallen, wird mit einer relativ geringen Schrittweite von zwei Tagen gerechnet. Im anderen Fall, bei sehr flachem Verlauf, wird Rechenzeit durch eine Verdopplung der Schrittweite eingespart. Die vorgesehene Laufzeitoptimierung erstreckt sich natürlich auch auf den Programmcode. Zeitintensive Vorgänge, wie z.B. die Ausgabe der Grafik auf dem Bildschirm, sollen durch das Einbinden schneller Maschinenprogramme in das (in einer höheren Programmiersprache geschriebene) Computerprogramm erheblich beschleunigt werden. Mit den genannten Maßnahmen konnte die für ein Modelljahr erforderliche Zeit schließlich auf ca. 4½ - 5½ Minuten, abhängig von der "Dynamik" im Jahreslauf, verkürzt werden.

Das Modell ist also in der Hauptsache einschichtig. Es erweist sich aber als notwendig, während der Sommerstagnation ein sehr einfaches Submodell der Tiefenschicht (Hypolimnion) mitzuführen, um die unterschiedlichen Veränderungen der Orthophosphatkonzentration in Epilimnion und Hypolimnion berücksichtigen zu können. Beide Schichten werden als homogen durchmischt angenommen.

In den folgenden Abschnitten 5.4.2. - 5.4.9. wird nun die Erstellung des mathematischen Modells erläutert. Hierzu sei generell angemerkt, daß mit den verschiedenen Hinweisen auf die Grenzen des Modells beabsichtigt ist, seine Leistungsfähigkeit durchsichtig zu machen. Die didaktischen Zielsetzungen des Computermodells werden durch die genannten Einschränkungen jedoch in keinem Fall beeinträchtigt.

5.4.2. Morphometrische und hydrologische Eigenschaften des Modellsees

THIENEMANN hat in seiner Pionierarbeit (THIENEMANN 1927) gezeigt, daß die morphometrischen Eigenschaften eines Sees, insbesondere die mittlere Tiefe z , für seine Produktivität in Relation zu den Auswirkungen klimatischer und edaphischer Faktoren von hoher Bedeutung sind. Bei der Festlegung der morphometrischen Daten des Modells *LAGO* ist also gezielt darauf zu achten, daß Werte gewählt werden, die das im Wortmodell beschriebene Modellverhalten ermöglichen und den erwünschten Seentyp repräsentieren. Spezielle Überlegungen sind vor allem zur Form des Seebeckens, zur mittleren Tiefe, zur maximalen Tiefe und zur relativen Tiefe notwendig.

Form des Seebeckens

Im Modell *LAGO* wird die Form des Seebeckens mit einem extremen elliptischen Sinusoid (vgl. Abb. 2) approximiert. NEUMANN (1959) hat in einer Untersuchung gezeigt, daß dieser geometrische Körper als mittlere Seenform gelten kann und damit ein "idealisiertes Seebecken" repräsentiert. Länge (2 km) und Breite (1,3 km) des Sees sind so gewählt, daß ein überschaubares Umfeld (1 Zufluß, 1 Abfluß, keine Gliederung des Sees in Teilbecken) angenommen werden kann.

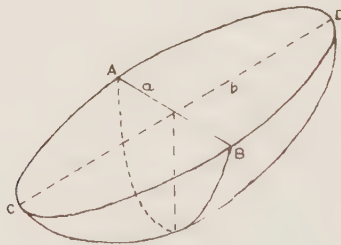


Abbildung 2: Idealisiertes Seebecken (extrem elliptisches Sinusoid). Die Halbachsen stehen im Verhältnis $a:b = 2:3$. Der Schnitt C-D hat die gleiche Form wie der Schnitts A-B. (entnommen aus STRAßERBA/GRAUCK, 1985, S.83, Fig. 4.3)

Das gesamte Wasservolumen des Seebeckens mit der Form eines extrem elliptischen Sinusoids bestimmt sich nach der Formel (vgl. WETZEL 1975, S. 34):

$$V = 4 (1 - 2/\pi) a b z_m \quad \text{bzw.} \quad V = 1.456 a b z_m$$

V ist das Volumen des Seebeckens, a und b sind die Halbachsen der elliptischen Oberfläche und z_m ist die maximale Tiefe des Sees.

Zur Berechnung der Volumina von Epilimnion und Hypolimnion in Abhängigkeit von der variablen Durchmischungstiefe ZMIXD (vgl. Abschnitt 5.4.4., S. 67) wurde die Transformation

$$p = 6 (z / z_m) - 3$$

verwendet (z = mittlere Tiefe, z_m = maximale Tiefe) und die relativen Volumina V_z in der normierten Tiefe

$$z' = ZMIXD / z_m$$

nach folgender Gleichungen bestimmt (STRASKRABA 1980, S. 54):

$$V_z = \{ 6 z' - 3 (1 - p) z'^2 - 2 p z'^3 \} / (3 + p).$$

Mittlere Tiefe (z)

Die mittlere Tiefe bestimmt sich aus dem Quotient von Seevolumen und Seeoberfläche. Für den Modellsee errechnet sich hierfür ein Wert von $z = 11,6$. Bei einem extremen elliptischen Sinusoid beträgt das Verhältnis $z : z_m$ immer 0.464, ein Wert, der dem Durchschnittswert von 0.467, den ANDERSON (1961) bei einem Vergleich von über 100 Seen für dieses Verhältnis erhalten hat, sehr nahe kommt. Die mittlere Tiefe wird als der aussagekräftigste Einzelwert für die morphometrischen Eigenschaften eines Sees angesehen. Sie zeigt bei großen Seen allgemein eine inverse Korrelation zur Produktivität auf allen trophischen Stufen. Die mittlere Tiefe ist somit auch für den Sauerstoffhaushalt eines Sees von Bedeutung, da die maximale Tiefe keine Rolle spielt, wenn das zugehörige Areal relativ klein ist.

Maximale Tiefe (z_m)

Der Modellsee *LAGO* soll auf die geographischen und klimatischen Verhältnisse eines Sees in der gemäßigten Klimazone zugeschnitten sein. Die Seen sind in diesem Bereich im allgemeinen dimiktisch. Bei der Festlegung der maximalen Tiefe des Modellsees muß also ein Wert gewählt werden, bei dem sich eine, je nach ausgewählter Temperaturkurve (vgl. Abschnitt 5.4.4., S. 67), mehr oder weniger intensiv ausgeprägte Schichtung im Sommer und Winter ausbilden kann.

Die maximale Tiefe darf aber auch nicht zu groß gewählt werden. Je größer der Wasserinhalt eines Sees ist, desto größer ist sein Vermögen, Wärmeenergie zu speichern. Das bedeutet, daß seine mittlere Temperatur den Änderungen der Lufttemperatur immer gedämpfter folgt. Je mehr der Anstieg des Volumens in einer Reihe von Seen durch eine Zunahme der Tiefe zustande kommt, desto größer ist die Ausdehnung desjenigen Tiefenbereichs, in welchem die Temperatur über das ganze Jahr hinweg niedrig und nahezu konstant bleibt (thermische Eigenschaften des Sees). Beim Modellsee soll diese Dämpfung nicht zu stark sein, damit die Auswirkungen unterschiedlicher klimatischer Verhältnisse (warmes - normales - kühles Jahr) auf die biologischen Komponenten im See spürbar werden.

Auch die Absicht, am Modellsee den Prozeß der anthropogen bedingten Eutrophierung aufzuzeigen, verlangt eine Begrenzung der maximalen Tiefe. Je tiefer der See, desto kleiner ist der Volumenanteil der trophogenen Zone, in welcher durch die photosynthetisch nutzbare Strahlung Pflanzensubstanz produziert wird und desto größer ist der Volumenanteil der tropholytischen Zone, in welcher die Abbauvorgänge vorherrschen.

Aufgrund der gesamten Überlegungen wurde für den Modellsee eine maximale Tiefe von 25 m festgelegt.

Relative Tiefe (z_r)

Der Modellsee soll im oligotrophen bis leicht eutrophen Zustand holomiktisch sein. Die Wasseroberfläche darf daher im Verhältnis zur Tiefe nicht zu klein sein. Da mit zunehmender Tiefe auch die Ausdehnung der Tiefenzone wächst, die bei kleiner Seeoberfläche durch windbedingte Durchmischungsvorgänge nicht mehr nennenswert beeinflußt wird, könnte sich ansonsten eine morphologisch bedingte Meromixis einstellen.

Das Verhältnis der Tiefe zur Seeoberfläche wird durch die relative Tiefe z_r ausgedrückt (WETZEL 1975, S. 31):

$$z_r = \frac{50 \ z_m \ \sqrt{\pi}}{\sqrt{A}}$$

z_m : maximale Tiefe

A: Oberfläche des Sees

Aus den bisher festgelegten morphometrischen Daten maximale Tiefe $z_m = 25\text{m}$, Länge = 2 km und Breite = 1,33 km errechnet sich ein Wert von $z_r \approx 1,53$. Es sind also zwei Voraussetzungen für das erwünschte Modellverhalten erfüllt: Einerseits ist der See so tief, daß sich eine bei günstigen Temperaturbedingungen deutlich ausgeprägte Schichtung (Sommerstagnation) einstellen kann, andererseits ist die Wasseroberfläche im Verhältnis zur Tiefe so groß, daß bei günstigen Windbedingungen im Frühjahr und Herbst eine Vollzirkulation möglich ist.

Zusätzlich sei noch angemerkt, daß die erwünschte Holomixis durch zwei weitere Bedingungen verhindert sein könnte, nämlich durch

- a) eine zu windgeschützte Lage, die eine topographisch bedingte Meromixis nach sich ziehen könnte,
- b) ein zu salzreiches Hypolimnion, was eine chemisch bedingte Meromixis zur Folge hätte (z.B. Toplitzsee).

Die Situation a) läßt sich dadurch ausschalten, daß der Modellsee unter der Annahme einer windexponierten Lage konzipiert wird. Was den Salzgehalt b) anbetrifft, so wird im Modell nur das gelöste Orthophosphat berücksichtigt. Selbst bei einem stark eutrophen See reicht die Orthophosphatkonzentration in der Regel alleine nicht aus, einen meromiktischen Zustand auszulösen. Es soll angenommen werden, daß auch die Konzentration anderer Salze den kritischen Wert nicht überschreitet.

Hydrologische Bilanz

Die Seen Mitteleuropas liegen in einer Region mit ausgeglichener hydrologischer Bilanz, d.h. die positiven Posten (Niederschläge, Kondensation auf der Oberfläche, horizontaler Zufluß) und die negativen Posten (Evaporation bzw. Evapotranspiration, horizontaler Abfluß) der hydrologischen Bilanzgleichung sind etwa gleich groß. Der Modellsees LAGO berücksichtigt nur die Größen Zu- und Abfluß, wobei das Zu- und Abflußvolumen als stets gleich groß festgelegt wird, andernfalls müßte ein sich im Jahreslauf veränderndes Wasservolumen zusätzlich berücksichtigt werden. Die restlichen Posten der hydrologischen Bilanz werden vernachlässigt.

Bei der Festlegung der Größen Zufluß und Abfluß muß beachtet werden, daß diese nicht nur für den Wasserhaushalt, sondern auch für den Stoffhaushalt von Bedeutung sind:

1. Je größer der Anteil der Zuflüsse gegenüber dem Niederschlag auf die Seeoberfläche ist, desto stärker kommen die chemischen Eigenschaften des zufließenden Wassers zum Tragen. Bei gleichen Niederschlägen pro m^2 nimmt der Einfluß des Einzugsgebietes auf den See zu, wenn der relative Anteil der Seeoberfläche abnimmt. Das Verhältnis dieser beiden Größen wird Umgebungsfaktor genannt (f_u). Für den Modellsee wird festgelegt:

$$f_u = \frac{\text{Fläche des Einzugsgebietes}}{\text{Fläche des Sees}} = \frac{50 \text{ km}^2}{2,1 \text{ km}^2} = 23,8$$

Die Ausgangssituation des Modellsees wird aufgrund dieses Umgebungsfaktors überwiegend von der chemischen Zusammensetzung des Zuflusses bestimmt. Die Fläche des Einzugsgebiets ist außerdem für die Höhe des Phosphatimports aus Niederschlägen ausschlaggebend (vgl. S. 75).

2. Die Veränderungen der chemischen Eigenschaften der Zuflüsse fallen umso mehr ins Gewicht, je langsamer der Wasseraustausch durch Zufluß und Abfluß erfolgt. Die Wasseraustauschzeit (oder Wassererneuerungszeit) T_w hängt vom Seevolumen (V) und vom Abflußvolumen / Zeit ab (Q_{DOUT}).

$$T_w = V / Q_{DOUT} \quad [d]$$

Die Wassererneuerungszeit ist ein theoretischer Mindestwert, der angibt, in welcher Zeitspanne das leere Seebecken aufgefüllt wäre. In der Realität werden aber die vorhandenen Wassermassen ausgetauscht, wobei je nach der Einschichtungstiefe des zufließenden Wassers unterschiedliche Zeiten beansprucht werden.

Das Modell *LAGO* soll einen See mit einer relativ langen Wassererneuerungszeit repräsentieren. Seevolumen ($V = 24,3$ Mill. m^3) und Abflußvolumen ($Q_{DOUT} = 33704$ m^3/d) wurden so gewählt, daß sich ein Wert von $T_w = 720$ Tagen ergibt. Es ist nicht möglich, diesen Wert im Modell zu verändern, da das mathematische Modell auf den Verhältnissen in einem See mit relativ langer Wasseraustauschzeit basiert. Für kleine Werte von T_w müßten andere mathematische Zusammenhänge modelliert werden.

Mit der Festlegung der genannten hydrologischen Daten sind natürlich nur wenige, stark vereinfachte Zusammenhänge angesprochen. In der Realität sind die stofflichen bzw. energetischen Wechselwirkungen zwischen einem Gewässer und seinem Einzugsgebiet sehr komplex. Auch kann in dem einfachen Modell keine Berücksichtigung der thermisch bedingten Zirkulationen und Konvektionsströmungen sowie der periodischen und unperiodischen Wasserbewegungen (z.B. fortschreitende Oberflächenwellen, Brandung, Seiches) erfolgen.

Da, wie bereits ausgeführt wurde, das Computermodell *LAGO* Allgemeingültigkeitscharakter besitzen soll, ist es nicht beabsichtigt, das Modellverhalten auf

einen real existierenden See abzustimmen. Trotzdem ist bei der Modellierung zur Orientierung die Fragestellung von Interesse, wie der Modellsee nach der Festlegung der morphometrischen und hydrologischen Eigenschaften in die Reihe einiger Seen im süddeutschen und österreichischen Raum einzuordnen wäre. Dies soll anhand folgender Tabelle gezeigt werden. Die Daten, die denen des Modellsees am nächsten kommen, sind unterstrichen (Daten aus SIEBECK 1982, S. 29-33).

	E	A	V	z_m	z	z_r	T_w	f_u
Modellsee	50	2,1	24,3	25,0	11,6	1,5	2	23,8
Waginger See	123,5	8,9	157	<u>27,0</u>	15,6	0,8		12,9
Kochelsee	683,7	5,9	183	66,2	31,3	2,4	0,12	114,9
Wörthsee	27,2	4,5	<u>17</u>	33,0	13,5	<u>1,4</u>		5,0
Ostersee	30,6	1,2	14	29,8	<u>11,7</u>	2,4		<u>24,5</u>
Klostersee		0,46	4	15,4	8	0,2		
Weissensee	14,6	1,4	<u>17</u>	<u>24,7</u>	<u>12,2</u>	1,9	1,25	9,4
Schliersee	27,1	<u>2,2</u>	53	40,4	23,9	2,4		11,3
Grüntensee		<u>2,5</u>	16	15,0	6,4	0,8		
Faakersee	<u>35,6</u>	<u>2,2</u>	33	29,5	14,9	1,8	1,2	15,2
Zeller See	27,5	3,47	53	32	15,3	<u>1,5</u>	<u>1,7</u>	6,9

E = Fläche des Einzugsgebiets [km^2]

A = Oberfläche des Sees [km^2]

V = Seevolumen [$10^6 \cdot \text{m}^3$]

z_m = maximale Tiefe [m]

z = mittlere Tiefe [m]

z_r = relative Tiefe [m]

T_w = Wassererneuerungszeit [Jahre]

f_u = Umgebungsfaktor [dimensionlos]

Tabelle 1: Morphologische Daten ausgewählter Seen

Mit den Größen Wassererneuerungszeit (T_w) und mittlere Tiefe (z) lassen sich die kritischen Belastungsgrenzen für den Phosphatimport abschätzen. Für den Modellsee ergibt der Quotient z/T_w den Wert 5,8. Legt man das VOLLENWEIDER-Diagramm (vgl. SIEBECK 1982, S. 80) zugrunde, liegt die kritische Grenze bereits bei ca. 0,6 mg P/ $\text{m}^2 \cdot \text{Jahr}$. Auf die Fläche des Modellsees umgerechnet neigt der See bereits bei einem Phosphatimport größer als 126 mg P/Jahr zur Eutrophierung.

5.4.3. Licht

Bei der Modellierung des exogenen Faktors "Licht" wird ein Submodell entwickelt, das, ausgehend von der auf die Seeoberfläche auftreffenden Globalstrahlung, den durch Reflexion, Absorption und Streuung verminderten, tiefenverteilten, photosynthetisch wirksamen Anteil ermitteln soll. Dieses Submodell steuert

- a) die Bruttophotosyntheserate und über sie die Respirationsrate,
- b) über die Tageslänge die Grazing-Rate des Zooplanktons.

(vgl. Darstellung in Anhang 1, S. 133)

Auf dem Bildschirm soll der jahreszeitliche Verlauf sowohl der auf die Seeoberfläche auftreffenden Strahlung als auch des Lichtanteils in 5m Wassertiefe dargestellt werden. Die Berechnung der Globalstrahlung lehnt sich an das von STRASKRABA (1980, S. 18-28) veröffentlichte Modell an, mit dem sich sowohl die geographische Lage als auch der Einfluß unterschiedlicher Bewölkung berücksichtigen läßt. Aus dieser Veröffentlichung wird außerdem eine mathematische Beziehung verwendet, mit der sich die Tageslänge ermitteln läßt (ebda. S. 27). Im folgenden Abschnitt werden die für das Submodell *Licht* erforderlichen Modellierungsschritte ausführlich erläutert.

1. Berechnung der Tageslänge (Photoperiode)

Die Tageslänge (PHOTPER, Lichtstunden pro Tag) wird im Modell bei der Berechnung der Globalstrahlung I und der Grazing-Rate des Zooplanktons bei Licht benötigt und durch die Cosinusfunktion

$$\text{PHOTPER} = 12 - 4 \cdot \cos t \quad [\text{h}]$$

approximiert (t: Zeit).

2. Berechnung der auf die Wasseroberfläche auftreffenden Globalstrahlung I

Die auf die Wasseroberfläche auftreffende Strahlung I wird mit der Sinusfunktion

$$I = (k_I \cdot 8,0 \cdot ((280 + 220 \cdot \sin(t + 280)) / 24) \cdot \text{photper } [\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}]$$

ermittelt. I ist hierbei der Momentanwert der auf eine horizontale Einheit der Atmosphärenoberfläche auftreffenden Strahlung (Einheit: $\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}$). Mit dem Koeffizienten k_I kann die Sinusfunktion an die zur Wahl stehenden unterschiedlichen klimatischen Bedingungen wie folgt angepaßt werden:

warmer Sommer/warmer Winter/wenig Bewölkung:	$k_I = 1,3$
normaler Sommer/normaler Winter:	$k_I = 1,0$
kühler Sommer/kalter Winter/häufige Bewölkung:	$k_I = 0,7$

Von der auf die Oberfläche eines Gewässers auftreffenden Gesamtstrahlung wird im Modell *LAGO* nur das sichtbare Licht (Wellenlänge 400-700 nm) berücksichtigt.

3. Berechnung des Lichts I_0 direkt unter der Wasseroberfläche

Bei der weiteren Erstellung des Submodells LICHT muß berücksichtigt werden, daß von der auf die Wasseroberfläche einfallenden direkten und diffusen Strahlung ein bestimmter, tages- und jahreszeitlich unterschiedlicher Anteil von der Oberfläche reflektiert wird. Liegt im Winter eine Schneedecke auf dem zugefrorenen See, so wird der Zutritt des Lichtes in den Wasserkörper weitgehend verhindert. Die jahreszeitlich unterschiedlichen Reflexionsverhältnisse werden durch die Beziehung

$$I_0 = R_I \cdot I \quad [\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}]$$

berücksichtigt. R_I stellt hierbei einen Reflexionsfaktor dar. Für die Zeit außerhalb der Winterstagnation erhält er den Wert 0.9, während der Winterstagnation, solange eine Eisschicht mit Schneebedeckung vorhanden ist, den Wert 0.2. Durch die Berücksichtigung der bei Eis und Schnee erheblich veränderten Albedo soll sich nach einem kalten Winter ein verzögerter Beginn des Phytoplanktonwachstums ergeben.

Nicht berücksichtigt wird bei dieser einfachen mathematischen Beziehung natürlich die Abhängigkeit der Albedo von der Strahlung, besonders vom Zenitwinkel (Sonnenhöhe) und vom Anteil der mit der Bewölkung im Zusammenhang stehenden diffusen und direkten Komponenten. Auch die Reflexionsfähigkeit der Wasseroberfläche, die vom Wellenschlag sowie durch Farbe und Trübung beeinflusst wird, bleibt außer Ansatz.

4. Berechnung der Transmission

Die durch Reflexion reduzierte Lichtmenge wird beim Durchgang durch die Wasserschichten selektiv zerstreut und absorbiert. Im Submodell LICHT muß daher die exponentielle vertikale Extinktion berechnet und die Reststrahlung I_{0z} (Transmission) in den jeweiligen Tiefen des Wasserkörpers ermittelt werden.

Vereinfacht kann die Strahlung I_λ mit der exponentiellen Abklingkurve (Lambert-Beersches Gesetz) approximiert werden:

$$dI_\lambda / dz = \varepsilon_\lambda \cdot z$$

bzw.

$$I_\lambda = I_0 \cdot e^{-\varepsilon_\lambda \cdot z}$$

I_λ ist hierbei die Lichtintensität einer bestimmten Wellenlänge λ in der Tiefe z , I_0 die Lichtintensität direkt unter der Wasseroberfläche, ε_λ der Extinktionskoeffizient.

Der homogene Tiefenabfall nach dem Lambert-Beerschen Gesetz gilt nur unter der Voraussetzung einer homogenen Verteilung der einzelnen lichtabsorbierenden Inhaltsstoffe (gelöste Stoffe, mineralische Trüb- und Schwebstoffe, Plankton, Tripton) und ist daher nur auf stark durchmischte und trübe Flachgewässer anwendbar. Wenn keine homogene Verteilung vorliegt, muß das Integral durch Summierung der Extinktion für jede Tiefenschicht (mit unterschiedlicher Dichte der oben genannten Inhaltsstoffe) ersetzt werden. Bei der Erstellung des Modells LAGO wurde zunächst die Reststrahlung I_{0z} in dieser Weise berechnet. Als Inhaltsstoff wurde das Phytoplankton berücksichtigt, wobei die Primärproduktion dann ebenfalls getrennt nach Tiefenschichten ermittelt wer-

den mußte. Es zeigte sich, daß dafür aber sehr hohe Rechenzeiten entstehen, so daß das Modell letztlich unter der einschränkenden Annahme der homogenen Verteilung vereinfacht wurde.

Für die Verringerung der Lichtintensität unter Wasser ist nicht nur die selektive Absorption durch gelöste Stoffe, sondern auch die weit weniger selektive Streuung durch Partikel maßgebend. Die Verluste durch Streuung sind beachtlich, wenn die Zuflüsse eines Standgewässers stark getrübt sind, z.B. infolge Erosion im Einzugsgebiet. Die mineralischen Trüb- und Schwebstoffe stellen wegen ihrer Konzentrationsschwankungen sowie ihrer spezifischen Extinktionsfaktoren (optische Eigenschaften des Materials, Größe und Form der Partikel) einen stark veränderlichen Lichtextinktionsfaktor dar. Das Modell berücksichtigt deshalb einen erhöhten Trübstoffeintrag durch die Schneeschmelze im Frühjahr und durch Starkregenereignisse. Die jahreszeitliche Dynamik wird hierbei in Anlehnung an BENNDORF (1979) durch die folgende trigonometrische Funktion approximiert:

$$FTE = 5 \cdot [1,5 + 0,5 \cdot \cos(t - 30)] \quad [\text{dimensionslos}]$$

FTE: Faktor für Trübstoffeintrag

t: Zeit

Mit der nun folgenden Fallunterscheidung wird unter Verwendung des Faktor FTE der Logarithmus der prozentualen Lichttransmission bei 5m Schichtdicke (LTE) ermittelt: (LTE wird bei der Berechnung des Extinktionskoeffizienten ϵ benötigt.)

$$LTE = 1,2 \quad \text{falls } FTE \leq FTE_{MAX}$$

$$LTE = 1,2 - [(FTE - FTE_{MAX}) / 2] \quad \text{falls } FTE > FTE_{MAX}$$

FTE_{MAX} ist hierbei der Wert von FTE, bei dessen Überschreiten eine Abnahme der Lichttransmission durch Trübstoffeintrag einsetzt. Für FTE_{MAX} wurde ein Wert von 4,3 gefunden. Der Maximalwert LTE = 1,2 tritt bei völligem Fehlen von Phytoplankton und anderen Trübstoffen auf.

Das Modell soll auch berücksichtigen, daß eine Massenentwicklung des Phytoplanktons den Streulichtanteil und damit die Lichtschwächung erhöht. Diese

rückkoppelnde Wirkung der Selbstbeschattung der Algen auf die Primärproduktion wird mit dem Produkt $X_{EXT} \cdot X$ erfaßt, wobei X_{EXT} die Konstante für die Lichtextinktion durch das Phytoplankton und X die Biomasse des Phytoplanktons darstellen. Mit einem Wert für X_{EXT} von 0,2 kann ein plausibles Modellverhalten erreicht werden.

Mit Hilfe der ermittelten Größen kann nun der vertikale Extinktionskoeffizient ϵ errechnet werden. Er setzt sich im Modell zu einem konstanten Anteil aus der Extinktion durch das reine Wasser und gelöste Substanzen (EPSAQUA) und zu variablen Anteilen aus der Lichtabnahme durch mineralische Trübstoffe sowie das Phytoplankton zusammen:

$$\epsilon = \text{EPSAQUA} - (\text{LTE} - X_{EXT} \cdot X) / 2 \quad [\text{dimensionslos}]$$

Für EPSAQUA wurde ein Wert von 0,95 verwendet.

Bei der Bemessung dieser Konstanten werden als Zielvorstellung die bei STRASKRABA/GNAUCK (1985, S.99 table 4.6) veröffentlichten Werte für ϵ zugrunde gelegt.

Die Reststrahlung I_{0z} in der durchmischten Schicht errechnet sich schließlich mit:

$$I_{0z} = \frac{\ln(K_1 + I_0) - \ln\{ K_1 + [I_0 \cdot \exp(-\epsilon \cdot Z_{MIXD})] \}}{\epsilon \cdot Z_{MIXD}} \quad [J \cdot cm^{-2} \cdot d^{-1}]$$

Hierbei ist

K_1 : Halbsättigungskonstante für die Beziehung zwischen Licht (400...700 nm) und Photosyntheserate $[J \cdot cm^{-2} \cdot d^{-1}]$

Z_{MIXD} : Tiefe der durchmischten Schicht

Bei der Modellierung wird die Tatsache nicht berücksichtigt, daß der Extinktionskoeffizient ϵ_{λ} stark wellenlängenabhängig ist. Je nach Spektralanteil dringt Licht gleicher Intensität unterschiedlich tief in das Wasser ein. Der im Modell errechnete Extinktionskoeffizient stellt nur einen Durchschnittswert für

Wellenlängen zwischen 400 und 700 nm dar. Gleiches gilt für spezielle gelöste Stoffe wie z.B. organische Fulvosäuren oder Huminsäuren.

Durch die Vernachlässigung der spektralen Zusammensetzung des Lichts kann daher auch nicht berücksichtigt werden, daß verschiedene Algenarten bzw. -Populationen auf Grund der Pigmentzusammensetzungen unterschiedliche Aktionsspektren haben (KIRK, 1975). Da das Modell *LAGO* aber nur eine Algenart enthält, fällt diese Einschränkung nicht weiter ins Gewicht.

5.4.4. Wassertemperatur

Obwohl die Temperatur des Wassers aufgrund seiner besonderen thermischen Eigenschaften weniger schwankt als die Lufttemperatur, stellt sie doch einen der wesentlichen limitierenden Faktoren dar, denn aquatische Organismen sind sehr häufig stenotherm. Auch im Computermodell *LAGO* soll die Bedeutung dieses abiotischen Faktors für verschiedene Prozesse im Ökosystem "See" deutlich werden. Aus der Darstellung in Anlage 1, S. 133 geht hervor, daß die Wassertemperatur auf zahlreiche Systemvariablen wirkt. Folgende Veränderungsraten werden von ihr beeinflußt:

- die Brutto-Photosyntheserate des Phytoplanktons,
- die Phytoplanktonrespiration,
- die Grazing-Rate des Zooplanktons,
- die Respirationsrate des Zooplanktons,
- die Mortalitätsrate des Zooplanktons.

Außerdem ist die Tiefenverteilung der Sauerstoff-Sättigungskurve von der Wassertemperatur abhängig.

Die Wassertemperatur (TEMP) wird im Modell durch eine Sinuskurve errechnet, die pro Zeitschritt einen Durchschnittswert für die 5m dicke Oberflächenschicht, in der sich die hauptsächliche Produktion abspielt, ermittelt:

$$\text{TEMP} = (\text{TK1} + \text{TK2}) + \text{TK1} \cdot \sin(t + 225)$$

Mit den Koeffizienten TK1 und TK2 wird die Sinuskurve an die zur Wahl stehenden klimatischen Bedingungen wie folgt angepaßt:

warmer Sommer/warmer Winter/wenig Bewölkung: TK1 = 10,5 TK2 = 1,5

normaler Sommer/normaler Winter: TK1 = 9,0 TK2 = 1,5

kühler Sommer/kalter Winter/häufige Bewölkung: TK1 = 8,0 TK2 = 0,5

5.4.5. Thermische Schichtung

Bei der Festlegung der morphometrischen und hydrologischen Daten des Modellsees (vgl. Abschnitt 5.4.2., S. 54) wurden die für eine idealisierte Abfolge von Zirkulation und Stagnation erforderlichen Grundvoraussetzungen bereits berücksichtigt: Zum einen darf das Gewässer nicht zu seicht sein, sonst sind die windbedingten Strömungen dominierend und es bildet sich kein charakteristisches Hypolimnion aus, zum anderen sollte die Verweilzeit des Wassers im Seebecken nicht zu kurz sein, sonst stehen die durch Zu- und Abfluß verursachten Durchmischungen im Vordergrund. Das festgelegte Verhältnis Volumen / Oberfläche des Modellsees und der Ausschluß einer topographisch- oder vegetationsbedingt windgeschützten Lage ermöglichen die Annahme einer mehreren Wochen dauernden, intensiven Frühjahrsvollzirkulation.

Als Ausgangspunkt für die Veranschaulichung der Abfolge von Zirkulation und Stagnation im Jahreslauf sowie der von den klimatischen Bedingungen abhängigen Intensität der thermischen Schichtung wurden zunächst für jede der drei festgelegten klimatischen Verhältnisse Temperatur-Isoplethen entworfen (vgl. Anlage 2, S. 135). Diese Isoplethen werden im Computerprogramm in Tabellen transformiert, aus denen sich dann während des Modellaufs die benötigten Werte für die vertikale Temperaturverteilung des Modellsees sowie die Startdekaden für die Zirkulations- und Stagnationsphasen entnehmen lassen.

Die Temperatur-Isoplethen wurden anhand veröffentlichter Daten realer Seen gewonnen ³⁰⁾ und idealisiert, damit sie eine generalisierende Aussage über die thermischen Verhältnisse von Seen (der gemäßigten Zone und einer dem Modellsee vergleichbaren Morphologie) abgeben können. Die drei Isoplethen-Darstellungen unterscheiden sich vor allem durch die mehr oder weniger deutliche Ausprägung der sommerlichen Schichtung, der damit verbundenen Tiefe der Durchmischung und der Dauer der Eis- und Schneedecke. Es wurde aber auch berücksichtigt, daß sich in einem "normalen" Jahr die Temperatur des tiefen Hypolimnions infolge der festgelegten maximalen Tiefe des Modellsees von 25 m kaum über 4 °C erhöht, während dies bei einem warmen Sommer durchaus der Fall sein kann (vgl. WETZEL 1975, S. 71). Als Ursache für diesen leichten

30) z.B. Temperatur-isoplethen des Starnberger Sees, in: Zur Limnologie des Starnberger Sees. Bayerisches Landesamt für Wasserwirtschaft (Hrsg.), München, 3/1982; Temperatur-isoplethen von Fischkaltersee, Weßlinger See, Obinger See in: Beiträge zur Limnologie bayerischer Seen. Bayerisches Landesamt für Wasserwirtschaft (Hrsg.), München 1/1982

Temperaturanstieg soll die Wärmeleitung und nicht die Sonneneinstrahlung gelten, da der Modellsee für letztere nicht die erforderliche hohe Transparenz vorsieht.

Durchmischungstiefe (ZMIXD)

Die Durchmischungstiefe beeinflusst die Dynamik des Modells erheblich. Von ihr sind abhängig (vgl. Darstellung in Anlage 1, S. 133):

- die Sedimentationsrate des Phytoplanktons,
- das Volumen von Epilimnion und Hypolimnion,
- davon wiederum die Orthophosphatkonzentration im Epilimnion und Hypolimnion, der Phosphatverbrauch durch Phytoplanktonwachstum sowie die Remineralisation des Orthophosphats.

Den größten Wert erreicht die Durchmischungstiefe ZMIXD während der Vollzirkulation im Frühjahr und Herbst. Sofern sich der See im holomiktischen Zustand befindet, ist ZMIXD gleich der maximalen Tiefe von $z_m = 25$ m. Die Werte in den übrigen Jahreszeiten hängen von der Stärke der sich jeweils ausprägenden Schichtung ab. In heißen Sommern stellt sich z.B. mit ZMIXD = 5 m der kleinste Wert ein.

Mit Hilfe einer mehrfarbigen grafischen Darstellung auf dem Bildschirm soll der Auf- und Abbau der thermischen Schichtung wie folgt veranschaulicht werden: Beim Fortschreiten des Frühlings wird die Oberflächenschicht schneller erwärmt, als sich die Wärme bei der Durchmischung verteilen kann. Ein Unterschied von nur wenigen Grad reicht aus, um die Frühjahrsvollzirkulation abzubrechen. Der Wasserkörper teilt sich dann in drei farblich voneinander abgesetzte Schichten auf, die sich, durch die kontinuierliche Zunahme der Temperatur der Oberflächenschicht, immer deutlicher ausbilden und gegeneinander sehr widerstandsfähig gegen Durchmischung sind (vgl. Farbfoto in Anlage 9, S. 264). Die jeweilige vertikale Temperaturverteilung ist durch sechs vertikal angeordnete Werte auf dem Bildschirm zu verfolgen. Die Intensität der Schichtung hängt dabei, wie bereits erwähnt, von den gewählten klimatischen Bedingungen ab.

Im Spätsommer und Herbst kühlt sich das Oberflächenwasser infolge fallender Lufttemperaturen ab, sinkt wegen der größeren Dichte nach unten und dringt

ins Metalimnion ein. Die Bildschirmgrafik zeigt dann, daß die Dicke der Schicht thermischer Diskontinuität abnimmt, während sie beim homothermen Epilimnion zunimmt. Wenn sich die Temperatur des gesamten Wasserkörpers auf ca. 4°C - 5°C vereinheitlicht hat, tritt der Modellsee unter der Annahme hoher Windgeschwindigkeiten (Herbststürme) in die Phase der Vollzirkulation. Die Durchmischung wird durch Pfeile gekennzeichnet (vgl. Anlage 6, Ausdruck 6, S. 168).

Solange sich im Winter keine Eisdecke auf dem See befindet, bleibt der Modellsee im Zustand der Vollzirkulation. Bei Ausbildung einer Eisdecke wird auf dem Bildschirm wiederum farblich gekennzeichnet, daß, da der Windeinfluß ausgeschaltet ist, die sogenannte "inverse Stratifikation" (kaltes Wasser liegt über wärmerem Wasser) auftritt. Der Dichteunterschied zwischen 4°C und 0°C ist sehr klein und verursacht nur einen kleinen Dichtegradienten. Das Wasser unter dem Eis stagniert nicht; die meisten Seen zeigen beträchtliche Wasserbewegungen unter dem Eis (WETZEL 1975, S. 74). Im Modell *LAGO* tritt eine Eisschicht immer zusammen mit einer Schneedecke auf. Damit ist der Fall ausgeschlossen, daß die Sonneneinstrahlung das Wasser unter dem Eis, das dann aber transparent sein muß, erwärmt.

Die Erwärmung des Wasserkörpers ist nach Analysen von Hutchinson (1941) und nachfolgender Arbeiten eine Kombination von direkter Sonneneinstrahlung, turbulenter Leitung und Dichteströmungen. Das Modell *LAGO* kann die komplizierten Zusammenhänge zwischen diesen drei Faktoren nicht wiedergeben. Es wird z.B. vereinfachend angenommen, daß die sich ausbildenden horizontalen Isothermen turbulente Strömungen so weit unterdrücken, daß diese vernachlässigt werden können. Das Modell ist aber in der Lage, den im Wortmodell vorgesehenen Zusammenhang zwischen Wassertemperatur, Durchmischungstiefe und Intensität der thermischen Schichtung sowie die Auswirkungen auf die Produktionsverhältnisse im Modellsee in vollem Umfang herzustellen.

Dateneingabe

Die Steuerung der Größen Licht, Tageslänge, Trübstoffeintrag und Wassertemperatur erfolgt, wie bereits aufgezeigt wurde, durch "harmonische Jahreskurven" in Form von trigonometrischen Funktionen. Grundsätzlich bestünde noch eine andere Möglichkeit: Die Eingangsgrößen werden vor dem Start oder während des Modellaufs als diskrete Werte eingegeben. Diese zweite Möglichkeit bietet den Vorteil, die Wirkung außergewöhnlicher meteorologischer und hydrologischer Ereignisse erfassen und untersuchen zu können, enthält aber zwangsläufig einen schwerwiegenden Nachteil für die praktische Arbeit: Wenn die Werte für die exogenen Größen in einem Zeitschritt von jeweils zehn Tagen einzugeben sind, fallen hierbei pro Simulationsjahr bereits weit über einhundert Einzelwerte an, die außerdem in einer Phase zeltaufwendiger Vorarbeit erst zusammengestellt werden müssen. Das Modell *LAGO* versucht, die Vorteile beider Möglichkeiten der Dateneingabe zu kombinieren: Werte, die für die Berechnung sehr häufig erforderlich sind, werden mittels harmonischer Jahreskurven erzeugt. Als Ausgleich dafür, daß die Abfolge der Werte im Jahreslauf fest vorgegeben ist, stehen dem Anwender drei verschiedenen Kurven zur Verfügung, die er zu Beginn eines Modelljahres jeweils neu auswählen kann. Werte, die nur einmal pro Simulationsjahr erforderlich sind, wie z.B. der im folgenden Abschnitt modellierte Phosphatimport, werden diskret eingegeben.

5.4.6. Phosphatkreislauf

Die Rolle der Nährstoffe im Stoffkreislauf wird im Modell *LAGO* exemplarisch am Phosphor aufgezeigt. Die Schlüsselfunktion dieses Pflanzennährstoffes beruht einerseits auf seinem geringen Angebot in naturbelassenen Gewässern, andererseits auf dem beträchtlichen Bedarf zum Aufbau pflanzlicher Biomasse. Phosphor ist als essentieller Nährstoff für die Primärproduzenten viel häufiger Minimumfaktor als Stickstoff und die Eutrophierung der Seen beruht in erster Linie auf der Zunahme der Phosphate (vgl. SCHWOERBEL 1984, S. 83 u. 103).

Das Gesamtphosphat in den Gewässern setzt sich aus den folgenden Fraktionen zusammen (nach WETZEL 1975, S. 232):

- a) organisches partikuläres Phosphat, das meist mehr als 95% des Gesamtphosphats im Wasser enthält;
- b) anorganisches, reaktiv gelöstes Phosphat als Orthophosphat (PO_4^{3-});
- c) organische Phosphorverbindungen mit niedrigem Molekulargewicht (bis ca. 250);
- d) lösliche hochmolekulare kolloidale Phosphorverbindungen mit einem Molekulargewicht größer als 5 Millionen.

Der Austausch erfolgt in der Hauptsache zwischen dem anorganischen und dem partikulären Phosphat (Phytoplankton, Bakterien). Von den Mikroorganismen wird ein Teil des Phosphats in Form der niedermolekularen organischen Phosphorverbindungen ausgeschieden. Letzere werden durch Polykondensation in die hochmolekularen Phosphorverbindungen umgewandelt. Beide Fraktionen, besonders aber die hochmolekulare, setzen lösliches, anorganisches Phosphat frei, das dann für eine rasche Aufnahme durch photoautotrophe Produzenten zur Verfügung steht. Das gelöste anorganische Phosphat wird dann wieder organisch gebunden und in Nahrungsketten eingeschleust. Im sommerlichen Epilimnion erfolgt dieser kurzgeschlossene Kreislauf ("turnover") des Phosphors innerhalb weniger Minuten.

Verluste aus dem Vorrat an partikulärem Phosphat entstehen durch Sedimentation aus der trophogenen in die tropholytische Zone und weiter bis zum Grund. Auf gleiche Weise entstehen auch Verluste über die Fraktion des kolloidalen Phosphats. Da die Phosphatspeicherung durch das Phytoplankton weit über den Eigenbedarf hinausgeht - es werden sozusagen Vorräte für Mangel-

zeiten angelegt - ist während eines starken Phytoplanktonwachstums im Wasser oft kein reaktives Phosphat nachweisbar. (vgl. SIEBECK 1982, S.62). Eine Nachlieferung ist durch die als Barriere wirkende Sprungschicht zwischen dem warmen Oberflächen- und dem kalten Tiefenwasser weitgehend unterbunden.

Das mit den organischen Resten im Sediment deponierte Phosphat wird unter *aeroben* Bedingungen an die Sedimentteilchen absorbiert oder als Eisenphosphat ausgefällt. Dieses Eisen(III)hydroxophosphat ist unlöslich, falls das Redoxpotential im Sediment größer als 0,2 Volt ist. Die oberste Sedimentschicht hat eine "Barriere"-Wirkung, solange noch einige mg O_2/l im darüberstehenden Wasser vorhanden sind; dann werden ausdiffundierte Ionen wieder ins Sediment zurückgerissen. Die Aufrechterhaltung einer oxidierten Grenzlamelle auf der Sedimentoberfläche benötigt viel weniger Sauerstoff als die Oxidierung einer reduktiven Grenzlamelle. Aber bereits bei weniger als 10% Sauerstoffsättigung setzt die Mobilisierung des Phosphors ein und bei erheblich weniger als 0,5 mg O_2/l kommt es zur fast explosionsartigen Freisetzung von PO_4^{3-} aus dem Sediment (FREVERT 1979). Nach SCHWOERBEL (1984, S. 85) kann diese "interne" Phosphorbelastung während der Sommermonate die gleiche Größe erreichen wie eine externe durch Zuflüsse und Abwässer.

Bei der Modellierung des Phosphathaushaltes sind nach STRASKRABA/GNAUCK (1985, S. 109) von den biologischen bzw. biochemischen Umwandlungsprozessen in der Reihenfolge die Nährstoffaufnahme, die Exkretion und die Autolyse am wichtigsten. Der erste erfolgreiche Versuch einer Nährstoffbilanzierung von Seen mittels steady-state-Modellen stammt von VOLLENWEIDER (1969). Diese Methode ist inzwischen zu einer Basis für die wissenschaftliche Behandlung der Eutrophierung geworden. BENNDORF (1980) hat das Vollenweider-Modell weiterentwickelt: Die Darstellung der biologischen Aufnahmemechanismen konnte vertieft und für den Phosphat-P angepaßt werden.

Einige Grundelemente dieses Modells von BENNDORF werden in das Modell LAGO übernommen. Dabei müssen aber, um die Rechenzeit niedrig zu halten, Vereinfachungen vorgenommen werden. Als Zustandsgröße ist im Modell nur das gelöste reaktive Orthophosphat in der durchmischten Schicht, nicht aber im Hypolimnion enthalten. Die Verhältnisse bei Sommerstagnation werden dafür folgendermaßen berücksichtigt: Zu Beginn der Sommerschichtung wird der aktu-

elle Wert der Orthophosphatkonzentration in einer Hilfsvariablen (PH) zwischengespeichert. Diese Hilfsvariable PH enthält damit die PO_4^{3-} -Konzentration im Hypolimnion. Sollte bei Vorliegen anaerober Verhältnisse am Seegrund eine Phosphatfreisetzung aus dem Sediment stattgefunden haben, so wird dies der zwischengespeicherten PO_4^{3-} -Konzentration des Hypolimnions zugeschlagen. Wenn die Herbstvollzirkulation einsetzt, werden der zuletzt aktuelle Wert für die durchmischte Schicht und der gespeicherte Wert für die Tiefenschicht verrechnet. Ein Phosphataustausch zwischen Epilimnion und Hypolimnion während der Sommerstagnation wird nicht modelliert, da er sich nicht spürbar auf das Modellverhalten auswirkt. Eine Erweiterung erhält das Modell *LAGO* durch die variable Durchmischungstiefe (vgl. Abschnitt 5.4.5., S. 69). Sie bringt den Vorteil, daß die Übergänge von der Zirkulations- zur Stagnationsphase und umgekehrt der Realität stärker angenähert und klimatische Einflüsse berücksichtigt werden können.

Die Orthophosphatbilanz der durchmischten Schicht wird nach folgender Gleichung ermittelt:

$$\text{dPD} = (\text{PDIMP} - \text{PDEXP} - \text{PDKONS} + \text{PDREMIN}) \quad [\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}]$$

Hierbei ist:

PD: Orthophosphatkonzentration in der durchmischten Schicht $[\text{mg P/m}^3]$

PDIMP: Phosphatimport in der durchmischten Schicht $[\text{mg P/d}]$

PDEXP: Phosphatexport aus der durchmischten Schicht $[\text{mg P/d}]$

PDKONS: Phosphatverbrauch durch Phytoplanktonwachstum in der durchmischten Schicht $[\text{mg P/d}]$

PDREMIN: Phosphatremineralisation in der durchmischten Schicht $[\text{mg P/d}]$

Auf den folgenden Seiten wird die Modellierung der einzelnen Bilanzposten erläutert.

Phosphatimport (PDIMP)

Das Modell *LAGO* berücksichtigt, daß Phosphate sowohl aus dem Einzugsgebiet als auch über die Niederschläge in den See gelangen können, wobei jeweils ein natürlicher und ein anthropogen bedingter Anteil zu unterscheiden ist. Die Größe des Phosphat-Imports kann für jedes Modelljahr über eine spezielle Bildschirmseite festgelegt werden (vgl. Anlage 5, Menü 5, S. 158). Von den vielen in Frage kommenden anthropogenen Belastungsquellen werden zwei typische ausgewählt: a) häusliche Abwässer, wobei bei der Berechnung die Verwendung phosphathaltiger Waschmittel eine Rolle spielt, und b) die Auswaschung gedüngter Wiesen und Ackerflächen. Zusätzlich kann die Art der Abwasserbehandlung festgelegt werden, wobei als Alternativen die direkte Einleitung des Abwassers in den See oder verschiedene Reinigungsstufen einer Kläranlage zur Wahl stehen. Bei der Festlegung des Phosphatimports aus Niederschlägen wird die Fläche des Einzugsgebiets (beim Modellsee 50 km²) berücksichtigt.

Wenn der Anwender die von ihm gewählten Daten eingegeben hat, erstellt das Modell auf dem Bildschirm eine Bilanz für den gesamten Phosphat-Import in g/Tag. Außerdem wird unter Berücksichtigung des Zuflußvolumens/Tag [m³/Tag] die Phosphat-Konzentration im Zufluß [mg P/m³] ermittelt und als Variable PIN an das Programm übergeben.

Der Phosphat-Import einer bestimmten Quelle wird anhand eines spezifischen Belastungskoeffizienten (SBK) und einer entsprechenden Einheit (z.B. Einwohnerzahl, Fläche) nach STRASKRABA/GNAUCK (1985, S. 103) mit der Gleichung berechnet:

$$P'_i = SBK_i \cdot E_i \text{ [g P/d]}$$

Hierbei ist:

P'_i : Fracht der i-ten Quelle

SBK_i : spezifischer Belastungsquotient der i-ten Quelle

E_i : Belastungseinheit der i-ten Quelle

Der Berechnung liegen folgende Werte zugrunde, die vom Anwender über eine weitere Bildschirmseite (Berechnungsgrundlagen, vgl. Anlage 5, Menü 6, S. 159) eingesehen werden können:

- a) Phosphat-Import aus Niederschlägen: 2,5 g/Tag (SBK₁)
- b) Phosphat-Import durch Abwässer aus Haushalten:
 - bei Verwendung phosphathaltiger Waschmittel: 2,3 g Phosphor/Tag (SBK₂)
pro Einwohner (E₂)
 - ohne Verwendung phosphathaltiger Waschmittel: 1,7 g Phosphor/Tag (SBK₃)
pro Einwohner (E₃)
- c) Phosphat-Import durch Auswaschung landwirtschaftlicher Flächen:
 - 3,8 g Phosphor/Tag (SBK₄)
pro ha (E₄)
- d) Die Abwasserbehandlung durch eine Kläranlage mit zusätzlicher chemischer Reinigungsstufe (D) vermindert die Phosphate um 90% . Die Alternativen A - C bewirken keine Reduktion der anfallenden Phosphatmenge.

Die spezifischen Belastungskoeffizienten (SBK) sind nach STRASKRABA/GNAUCK (1985, table 5.3, S. 104) festgelegt und auf g/Tag umgerechnet. Die Autoren weisen darauf hin, daß ihre aus verschiedenen Literaturangaben zusammengefaßten Mittelwerte von SBK eine hohe Streuung aufweisen. Zur Festlegung der Werte für das Modell LAGO werden deshalb weitere Angaben hinzugezogen. Nach WAGNER (1976) kann man für jeden Einwohner mit einer täglichen "Phosphatproduktion" von mindestens 3 g im häuslichen Abwasser rechnen, von der das meiste in die Gewässer gelangt. Hinzu kommt noch etwa 1 g/Einwohner aus überdüngten Böden und anderen Quellen. In der limnologischen Projektstudie über den Königsee (SIEBECK 1982, Tabelle 21, S. 78) werden die Phosphat-Importe durch Auswaschung mit 8 kg/km² pro Jahr, durch den Menschen mit 1,9 g P/Tag beziffert, wobei bei letzterem Wert noch ein Zuschlag von 20% für phosphathaltige Waschmittel hinzukommt.

Der für die Bilanzgleichung der Orthophosphatbilanz erforderliche Wert für den Phosphatimport in der durchmischten Schicht (PDIMP) wird schließlich wie folgt ermittelt:

$$\text{PDIMP} = \text{PIN} \cdot \text{QDIN} \quad [\text{mg P/d}]$$

Hierbei ist:

PIN: (Ortho-)Phosphatkonzentration im Zufluß $[\text{mg P/m}^3]$

QDIN: Wasserzufluß in der durchmischten Schicht $[\text{m}^3/\text{d}]$

Phosphat-Export

Das Modell *LAGO* setzt vereinfachend voraus, daß die Wassermengen von Zufluß und Abfluß gleich groß sind und der Abfluß nur aus der durchmischten Schicht erfolgt. Der Phosphat-Export aus der durchmischten Schicht (PDEXP) wird mit Hilfe der folgenden Gleichung ermittelt:

$$\text{PDEXP} = \text{PD} \cdot \text{QDOUT} \quad [\text{mg P/d}]$$

$$\text{QDOUT} = \text{QDIN}$$

Hierbei sind:

PD: Orthophosphatkonzentration in der durchmischten Schicht $[\text{mg P/m}^3]$

QDOUT: Wasserabfluß aus der durchmischten Schicht $[\text{m}^3/\text{d}]$

QDIN: Wasserzufluß in die durchmischte Schicht $[\text{m}^3/\text{d}]$

Phosphatverbrauch durch Phytoplanktonwachstum (PDKONS)

Zur Ermittlung des Phosphatverbrauchs durch Phytoplanktonwachstum in der durchmischten Schicht (PDKONS) benötigt man die Wachstumsrate XGR, um die Verknüpfung mit der Primärproduktion herzustellen:

$$\text{PDKONS} = \text{XGR/KXNUP} \cdot \text{X} \cdot \text{VOLD} \quad [\text{mg P/d}]$$

XGR: Wachstumsrate des Phytoplanktons in der durchmischten Schicht $[\text{d}^{-1}]$

X: Phytoplanktonbiomasse (-dichte) $[\text{cm}^3/\text{m}^3]$

VOLD: Volumen der durchmischten Schicht $[\text{m}^3]$

KXNUP: Phosphatausnutzungskoeffizient für Phytoplankton $[\text{cm}^3/\text{mg P}]$

Die Modellierung der Wachstumsrate XGR wird in Abschnitt 5.4.7. (S. 83) erläutert. Mit der Konstanten KXNUP könnte das Submodell auf bestimmte Algengruppen eingestellt werden. Bei der Standardeinstellung des Modells (Normallauf), bei dem die "Modell-Algen" miktoplanktische Diatomeen repräsentieren, wird für KXNUP ein Wert von 0,8 verwendet. Die Konstante kann vom Benutzer über das Menü "Modellexperimente" (vgl. Anlage 5, Menü 2, S. 155) verändert werden.

Phosphatremineralisation (PDREMIN)

Zur Phosphatremineralisation in der durchmischten Schicht (PDREMIN) trägt zum einen Teil die Remineralisationsrate aus abgestorbenem oder von Fischen gefressenem Zooplankton (PZMORT) und zum anderen Teil die Remineralisationsrate aus der Exkretion des Zooplanktons (PEXKR) bei. Z ist die Zooplanktonbiomasse (-dichte; Einheit: cm^3/m^3), VOLD das Volumen der durchmischten Schicht $[\text{m}^3]$:

$$\text{PDREMIN} = (\text{PZMORT} + \text{PEXKR}) \cdot Z \cdot \text{VOLD} \quad [\text{mg P/d}]$$

Die Rate der Phosphatremineralisation aus abgestorbenem oder von Fischen gefressenem Zooplankton (PZMORT) wird als fester Prozentsatz (50%) aus der Mortalitätsrate des Zooplanktons (ZMORA) gewonnen:

$$\text{PZMORT} = 0,5 \cdot \text{ZMORA} / \text{KZNUP} \quad [\text{mg P} \cdot \text{cm}^{-3} \cdot \text{d}^{-1}]$$

Mit dem Koeffizienten KZNUP (Einheit: $\text{cm}^3/\text{mg P}$) kann die Phosphatausnutzung für Zooplankton bestimmt werden. Im Modell wird ein Wert von 0,8 verwendet. Die Mortalitätsrate ZMORA wird in Abschnitt 5.4.8., S. 90 erläutert.

Die Rate der Phosphatremineralisation durch Exkretion des Zooplanktons (PEKKR) ergibt sich aus der Differenz der in den Verdauungstrakt aufgenommenen und der für das Wachstum verbrauchten Gesamtposphatmenge, wobei ein fester Prozentsatz (50%) der Ausscheidungen wieder in den kurzgeschlossenen Nährstoffkreislauf einfließt. Der Rest sedimentiert mit dem Kot.

$$\text{PEKKR} = 0,5 \cdot \left[\frac{\text{ZGZRA}}{\text{KKNUP}} - \frac{\text{ZGR}}{\text{KZNUP}} \right] \quad [\text{mg P} \cdot \text{cm}^{-3} \cdot \text{d}^{-1}]$$

ZGZRA: Grazing-Rate des Zooplankton [d^{-1}], vgl. S. 86

ZGR: Wachstumsrate des Zooplanktons [d^{-1}], vgl. S. 89

KKNUP: Phosphatausnutzungskoeffizient für Phytoplankton [$\text{cm}^3/\text{mg P}$]

KZNUP: Phosphatausnutzungskoeffizient für Zooplankton [$\text{cm}^3/\text{mg P}$]

Bei der Modellierung wird, wie bereits dargestellt, davon ausgegangen, daß der Hauptweg der Phosphatremineralisation direkt oder indirekt über das Zooplankton verläuft (Ausscheidung durch lebendes Zooplankton, Autolyse abgestorbenen Zooplanktons, Ausscheidung durch zooplanktonfressende Fische). Da die Phytoplanktonverluste durch Absterben (Parasitismus, Nährstoffmangel, Lichtmangel) im Modell nicht enthalten sind, muß auch die Phosphatremineralisation über die Bakterien unberücksichtigt bleiben.

Ebenfalls nicht berücksichtigt werden konnte die Tatsache, daß Planktonalgen bei ausreichender Sauerstoffversorgung mehr Phosphat aufnehmen als sie benötigen und den Überschuß als Polyphosphat speichern. "Mit derartigen Phosphatreserven ausgerüstete Algen können auch ohne Phosphat im Wasser weiterwachsen. Stehen keine internen P-Vorräte mehr zur Verfügung, so wird über eine adaptive Produktion des Exoenzyms Alkalische Phosphatase die Hydrolyse organischer P-Verbindungen katalysiert und Phosphate damit der autotrophen Biosynthese zugeführt; die Alkalische Phosphatase wird dann wieder 'abgeschaltet'." (SCHWOERBEL 1984, S. 104) Da nur eine Algengruppe in die Modellierung einbezogen wird, spielt die Vernachlässigung der in der Realität vorhandenen unterschiedlichen Affinität der Algen zum gelösten Orthophosphat bzw. spielen die divergierenden Aufnahmekinetiken, welche für die Konkurrenzphänomene bzw. für Verschiebungen in der Phytoplanktonzusammensetzung bei steigender Phosphatbelastung eines Gewässers entscheidend sind, aber keine Rolle.

Phosphataustausch zwischen Sediment und Freiwasser

Der Austausch zwischen Freiwasser und Sediment spielt im Phosphorhaushalt eines Sees eine bedeutsame Rolle. Einerseits wird meist der überwiegende Teil des in ein Gewässer eingetragenen Phosphors im Sediment stillgelegt (vgl. KLAPPER et. al., 1974, BURNS und ROSS, 1971), andererseits bildet die Phosphorfreisetzung aus dem Sediment eine wichtige Nährstoffzufuhr insbesondere in der zuflußarmen und wachstumsintensiven Sommerzeit. Diese Prozesse sind in der Realität sehr komplex. Die Größe der Phosphoreliminierung und der Phosphorrückführung wird z.B. durch eine Reihe von Vorgängen im Sediment (Abbau des organischen Materials, chemische Bindung, Adsorption) bestimmt und von vielen äußeren Faktoren beeinflusst (z.B. Temperatur, Sauerstoffgehalt, pH-Wert, Aktivität der Bodenorganismen).

Die Modellierung dieses Phosphataustausches bereitete einiges Kopfzerbrechen. Für eine exakte Quantifizierung wäre natürlich ein umfangreiches mathematisches Modell erforderlich. Hierfür stehen aber keine Rechenkapazitäten zur Verfügung. Ziel der Modellierung ist, wie im Wortmodell beschrieben, zu zeigen, daß unter anaeroben Bedingungen am Gewässergrund beträchtliche Mengen an löslichem Orthophosphat aus dem Sediment freigesetzt werden können und eine erhebliche "interne Düngung" des Sees verursachen. Dieser Prozeß kann mit Hilfe einer vergleichsweise sehr einfachen Gleichung, die KAMP-NIELSEN (1975) für den dänischen See ESROM verwendet hat, approximiert werden. Auch JÖRGENSEN et. al. (1975, S. 134) bezeichnen diese Gleichung als zufriedenstellende Annäherung an die beobachteten Freisetzungswerte. Der Phosphoraustausch zwischen Sediment und Freiwasser unter anaeroben Bedingungen wird hierbei als Diffusionsprozeß beschrieben:

$$F_{diff} = 1.21 (P_i - P_s) - 1.70$$

F_{diff} : Phosphat-Freisetzung aus dem Sediment [$\text{mg P} \cdot \text{m}^{-2} \cdot \text{d}^{-1}$]

P_i : (mit Ammoniumchlorid extrahierbare) Phosphatkonzentration im obersten Zentimeter des Interstitialwassers [mg P/l]

$P_i - P_s$: Konzentrations-Gradient über dem obersten Zentimeter der Sedimentschicht [mg P/l]

P_s : Konzentration des gelösten Phosphors im Freiwasser [mg P/l]

Die bei der Berechnung von F_{diff} notwendige Sedimentfläche läßt sich anhand der festgelegten Form des Seebeckens (vgl. Abschnitt 5.4.2., S. 54) ermitteln. Der Wert von P_s ist identisch mit der Orthophosphatkonzentration im Hypolimnion (PH). Wenn die Sauerstoffkonzentration am Gewässergrund kleiner als 0,5 mg O_2/l ist, dann wird pro Zeitschritt ein entsprechender Betrag von F_{diff} zur Orthophosphatkonzentration im Hypolimnion addiert.

Da, wie KAMP-NIELSEN (1975, S.154) anmerkt, der Diffusionsvorgang von der Wassertemperatur nur gering beeinflußt wird, wird auf eine diesbezügliche Korrektur der Gleichung verzichtet.

Die Verifikation der Phosphatfreisetzung aus dem Sediment erweist sich als schwierig. Da hierzu kaum Meßwerte vorliegen, kann diese Größe nur sehr grob geschätzt werden. BANOUB (1974) gab Freisetzungswerte zwischen 1,2 und 9,4 mg $\text{P/m}^2/\text{d}$ an. IMBODEN (1973) zitierte sogar Werte von 35 .. 500 mg $\text{P/m}^2/\text{d}$. Werden diese Angaben in Beziehung zu den Belastungswerten z.B. der Darßer Boddenkette von ca. 3 mg $\text{P/m}^2/\text{d}$ oder von der Wahnbachtalsperre (BERNHARDT 1973) von 3,8 mg $\text{P/m}^2/\text{d}$ gesetzt, dann ergibt sich, daß die Phosphorrückführung aus dem Sediment etwa so groß wie die Zufuhr von außen sein könnte. Dies deckt sich mit der bereits zitierten Aussage von SCHWOERBEL (S.73).

5.4.7. Primärproduktion

Die Primärproduktion beruht auf dem biochemischen Prozeß der Photosynthese, bei der Strahlungsenergie biochemisch gebunden wird. Im Gewässer sind hierzu in der Hauptsache das Phytoplankton, das Periphyton und die Makrophyten, aber auch Cyanobakterien sowie andere Bakterien mit Photosynthesefarbstoffen befähigt.

Mathematisch kann die Produktion als Biomasseänderung in einem Zeitabschnitt, also als Produktionsrate, erfaßt werden. Im Gewässer stehen der Produktion (PROD) ständig Verluste durch Respiration (RESP), Exkretion (EXKR), Tod (MORT), Fraß (GRAZE), Absinken (SINK) und Export (EXP) entgegen. Demzufolge ergibt sich für die Änderung der Biomasse (B):

$$dB/dt = B [(PROD - RESP) - (EXKR + MORT + GRAZE + SINK + EXP)]$$

Die individuelle Nettoproduktion (PROD - RESP) hängt ab von der Stoffaufnahme und Ausnutzung (bei Konsumenten Ingestion und Assimilation), sowie vom individuellen Metabolismus und ist eine Bilanz hauptsächlich aus diesen beiden Größen. Daraus ergibt sich der ökologische Spielraum für die Produktion der Organismen, da beide Größen in vielfältiger Beziehung zu den Umweltfaktoren stehen.

Im Modell *LAGO* wird der Vorgang der Primärproduktion exemplarisch an den Algen aufgezeigt. Sie sind die wichtigsten Produzenten in der Freiwasserzone. Die Produktion höherer Pflanzen ist nur im Litoral der Seen und in durchlichteten Flachgewässern von Bedeutung. Von den Algen kann aus Gründen der begrenzten Rechenkapazität der verwendeten Computer nur eine Klasse modelliert werden. Die "Modellalgen" sollen durch die Bacillariophyceae repräsentiert werden, da diese Klasse - bei Annahme idealisierter Verhältnisse - ein ausgeprägtes Frühjahrsmaximum zeigt und damit der Einfluß der abiotischen Faktoren Licht und Temperatur auf das Algenwachstum besonders deutlich gemacht werden kann.

Die sich bei der Primärproduktion einstellende Populationsänderung wird im Modell in Wachstumsprozesse (Phytoplanktonwachstum XG) und Verluste (Sedi-

mentation XSEDI, Zooplanktonfraß ZGZ) unterteilt. Es ergibt sich folgende Gleichung, deren einzelne Bestandteile im folgenden erläutert werden:

$$dX = XG - XSEDI - ZGZ \quad [\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}]$$

Phytoplanktonwachstum (XG)

Bei der Ermittlung der Primärproduktion ist zwischen der Bruttoproduktion, die mit der Photosynthese identisch ist, und der Nettoproduktion, die den reinen Biomassezuwachs darstellt, zu unterscheiden. Im Modell *LAGO* wird die Wachstumsrate XGR (Nettoproduktion) in der durchmischten Schicht aus der Differenz zwischen der Brutto-Photosyntheserate (XBPR) und der Respirationsrate (XRESP) ermittelt:

$$XGR = XBPR - XRESP \quad [\text{d}^{-1}]$$

Für die quantitative Beschreibung der Photosynthese werden bei STRASKRABA/GNAUCK (1985, S. 122-123) zwei Wege angegeben: die klassischen Photosynthese-Modelle (Lichtabhängigkeit der Photosynthese), die in den meisten bisher entwickelten Simulationsmodellen für wissenschaftliche Zwecke enthalten sind, sowie Modelle, die die Kopplung zahlreicher Mechanismen berücksichtigen (Kinetik der Mehrfachvorräte). Aus der letztgenannten Kategorie hat sich bei zahlreichen Vorversuchen für die Verwendung im Modell *LAGO* ein Submodell als geeignet erwiesen, das von BENNDORF (1979) verwendet wurde. Dieses Submodell berücksichtigt neben dem Faktor Strahlung weitere Umweltgrößen: Die Brutto-Photosyntheserate des Phytoplanktons in der durchmischten Schicht (XBPR) hängt hyperbolisch von der Strahlungsintensität und dem anorganischen gelösten Orthophosphat, linear von der Wassertemperatur sowie invers hyperbolisch von der Biomasse des Phytoplanktons ab:

$$XBPR = \left[XBPRMIN + \left(\frac{XBPRMAX - XBPRMIN}{XGTOPT} \right) \cdot TEMP \right] \cdot \frac{PD/X}{KXP/KX + PD/KX + KXP/X + PD/X} \cdot I_{0z} \quad [\text{d}^{-1}]$$

XBPRMAX: maximale Bruttophotosyntheserate [d^{-1}]
 XBPRMIN: Bruttophotosyntheserate bei Temperatur nahe 0°C und Optimalbedingungen bezüglich Licht und Phosphat [d^{-1}]
 XGTOPT: Optimaltemperatur für Phytoplanktonwachstum [$^{\circ}\text{C}$]
 I_{0z} : Licht (400...700 nm) in der durchmischten Schicht [$\text{J}\cdot\text{cm}^{-2}\cdot\text{d}^{-1}$]
 KX: Halbsättigungswert für die inverse Beziehung zwischen Photosynthese und Biomasse des Phytoplanktons [dimensionslos]
 KXP: Halbsättigungskonstante für die Beziehung zwischen Phosphat und Photosyntheserate (XBPR) bei minimaler Phytoplanktonbiomasse [mg P/m^3]
 PD: Orthophosphatkonzentration in der durchmischten Schicht [mg P/m^3]
 TEMP: Wassertemperatur [$^{\circ}\text{C}$]
 X: Phytoplanktonbiomasse bzw. -dichte [cm^3/m^3]

In obigem Submodell ist das bereits in Abschnitt 5.4.6., S. 73 erwähnte, von VOLLENWEIDER (1969) begründete und von BENNDORF (1980) weiterentwickelte Modell für die Nährstoffaufnahme enthalten, das durch den Faktor

$$\frac{\text{PD/X}}{\text{KXP/KX} + \text{PD/KX} + \text{KXP/X} + \text{PD/X}}$$

dargestellt wird. Es berücksichtigt die inverse Abhängigkeit der spezifischen Wachstumsrate von der Biomasse der Algen, wobei KX der Halbsättigungswert für diese Abhängigkeit und KXP die Halbsättigungskonstante der Phosphataufnahme einer bestimmten Phytoplanktonart, in diesem Fall der Bacillario-phyceae, ist.

Der Halbsättigungswert KX ermittelt sich aus

$$\text{KX} = \text{KXP1} \cdot \text{PD}^{\text{KXP2}} \quad [\text{dimensionlos}]$$

KXP1: Konstante der Abhängigkeit des Halbsättigungswertes KX von der Phosphatkonzentration [$\text{cm}^3/\text{mg P}$]

KXP2: Konstante der Abhängigkeit des Halbsättigungswertes KX von der Phosphatkonzentration [dimensionslos]

PD: Orthophosphatkonzentration in der durchmischten Schicht [mg P/m^3]

Allerdings muß die bei BENNDORF (1979) ursprünglich vorgenommene Berechnung der Photosyntheserate für mehrere Schichten des Wasserkörpers wegen

sehr hoher Rechenzeiten aufgegeben und durch ein Integral mit den Integrationsgrenzen Wasseroberfläche und Durchmischungstiefe (ZMIXD) ersetzt werden. Diese Maßnahme erfordert auch hier die vereinfachende Annahme einer regelmäßigen, auf die durchmischte Schicht beschränkte Vertikalverteilung des Phytoplanktons.

Bei der Modellierung der *Respirationsrate* $XRESP$ wird berücksichtigt, daß die Atmungsgeschwindigkeit mit der Intensität der Photosynthese und der Wachstumsgeschwindigkeit ansteigt. Es wird zwischen einem passiven, temperaturabhängigen Anteil XR_{PP} , der bei Lichtabwesenheit wirkt, und einen aktiven Anteil XR_{Pa} unterschieden. Nach BANNISTER (1979) besteht zwischen XR_{Pa} und der Wachstumsgeschwindigkeit der Algen, nach McCREE (1970) zwischen XR_{Pa} und der Photosynthese ein linearer Zusammenhang. Die für Photosynthese und Wachstum maßgeblichen Umweltbedingungen Licht, Temperatur und Vorhandensein von Nährstoffen gelten somit auch für die Respiration. Man erhält als mathematisches Modell für die Gesamtrespiration (vgl. STRASKRABA/GNAUCK 1985, S. 133):

$$XRESP = \underbrace{XRPP}_{\text{Grundumsatz}} + \underbrace{0,3 \cdot XBPR}_{\text{Leistungsumsatz}} \quad [d^{-1}]$$

$XBPR$: Brutto-Photosyntheserate des Phytoplanktons in der durchmischten Schicht [d^{-1}]

$XRPP$: Grundumsatz der Phytoplanktonrespiration [d^{-1}]

Der linear von der Wassertemperatur TEMP abhängige Grundumsatz der Phytoplanktonrespiration $XRPP$ errechnet sich nach:

$$XRPP = 0,02 \cdot TEMP \quad [d^{-1}]$$

Das Phytoplanktonwachstum XG ergibt sich abschließend durch:

$$XG = XGR \cdot X \quad [cm^3 \cdot m^{-3} \cdot d^{-1}]$$

X : Phytoplanktonbiomasse bzw. -dichte [cm^3/m^3]

XGR : Wachstumsrate des Phytoplanktons in der durchmischten Schicht [d^{-1}]

Nicht berücksichtigt wird bei der Modellierung der Respiration die mit der Ausscheidung organischer Stoffe eng gekoppelte *Photorespiration*.

Populationsverluste

Bei den Populationsverlusten sind interne Verluste und Verluste durch Transportprozesse zu unterscheiden. Als *interne Verluste* bezeichnet man solche durch natürliche, physiologische Mortalität und Fraß durch Konsumenten. Im Modell *LAGO* werden nur die Verluste durch Zooplanktonfraß modelliert. Hierfür wird wiederum ein Submodell nach BENNDORF (1979) verwendet. Bei diesem Submodell hängt die Grazing-Rate in Form einer Optimumkurve von der Temperatur, hyperbolisch von der Algenmenge sowie invers hyperbolisch von der Zooplanktonbiomasse ab. Außerdem wird die Lichthemmung der Zooplanktonfraßaktivität im sommerlichen Epilimnion berücksichtigt. Das Submodell setzt sich wie folgt zusammen:

1. Berechnung der Phytoplanktonverluste infolge Freßtätigkeit ZGZ (grazing) des Zooplanktons

$$ZGZ = ZGZRA \cdot Z \quad [\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}]$$

Z: Zooplanktonbiomasse bzw. -dichte [cm^3/m^3]

2. Berechnung der Grazing-Rate ZGZRA des Zooplanktons

Fallunterscheidung:

$$ZGZRA = ZGZDU \text{ falls } HVZ \leq t < SST \quad [\text{d}^{-1}]$$

$$ZGZRA = ZGZLI \text{ falls } SST \leq t < HVZ \quad [\text{d}^{-1}]$$

SST: Sommerstagnation

HVZ: Herbstvollzirkulation

3. Berechnung der Grazing-Rate des Zooplanktons bei Dunkelheit (ZGZDU)

$$\text{ZGZDU} = \left\{ 2,25 \cdot \left[\exp \left(-2,0 \cdot \text{abs} \left(\ln \frac{\text{TEMP}}{\text{ZGZTOPT}} \right) \right) \right] + 0,25 \right\} \\ \cdot \frac{\text{X/Z}}{\text{KGZX/KZ} + \text{X/KZ} + \text{KGZX/Z} + \text{X/Z}} \quad [\text{d}^{-1}]$$

$$\text{KZ} = \text{KZP1} \cdot \text{X}^{\text{KZP2}} \quad [\text{dimensionslos}]$$

- KGZX: Halbsättigungskonstante für die Beziehung zwischen Grazing-Rate und Phytoplanktonbiomasse [cm^3/m^3]
 KZ: Halbsättigungswert für die inverse Beziehung zwischen Grazing-Rate und Biomasse des Zooplanktons [dimensionslos]
 KZP1: Konstante der Abhängigkeit des Halbsättigungswertes KZ von der Phytoplanktonbiomasse [cm^3/cm^3]
 KZP2: Konstante der Abhängigkeit des Halbsättigungswertes KZ von der Phytoplanktonbiomasse [dimensionslos]
 TEMP: Wassertemperatur [$^{\circ}\text{C}$]
 X: Phytoplanktonbiomasse bzw. -dichte [cm^3/m^3]
 Z: Zooplanktonbiomasse bzw. -dichte [cm^3/m^3]
 ZGZTOPT: Optimaltemperatur für Grazing [$^{\circ}\text{C}$]

4. Berechnung der Grazing-Rate des Zooplanktons bei Licht (ZGZLI)

$$\text{ZGZLI} = \text{ZGZDU} \cdot \left(\frac{24 - \text{PHOTPER}}{24} \right) + \text{ZGZI} \cdot \text{ZGZDU} \cdot \frac{\text{PHOTPER}}{24} \quad [\text{d}^{-1}]$$

- PHOTPER: Photoperiode (Tageslänge) [h]
 ZGZI: Lichthemmungsfaktor für Grazing [dimensionslos]

Bei den Verlusten durch *Transportprozesse* wird nur die Sedimentation der Algen in die Modellierung einbezogen. Durch Import und Export verursachte Populationsänderungen werden vernachlässigt. Auch Verluste, die durch den gegenseitigen Austausch der Algen zwischen horizontalen benachbarten Schichten oder vertikalen Zonen verursacht werden, sind im Modell ausgespart.

Im allgemeinen sinken die Algen sehr langsam mit einer Geschwindigkeit, die sich aus dem Übergewicht und dem Formwiderstand der Zelle sowie den Wasserbewegungen (Zirkulations- und Turbulenzbewegungen) ergibt. Im Modell LAGO errechnet sich die Sedimentationsrate SEDRA aus dem Quotienten aus einem Festwert der Sinkgeschwindigkeit der Algen im unbewegten Wasser und der aktuellen Durchmischungstiefe. Somit wird die aufwärts gerichtete Wasserbewegung infolge Durchmischung näherungsweise berücksichtigt:

$$XSEDI = XSEDRA \cdot X \quad [\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}]$$

$$XSEDRA = XS / ZMIXD \quad [\text{d}^{-1}]$$

X: Phytoplanktonbiomasse bzw. -dichte $[\text{cm}^3/\text{m}^3]$

XS: Nettosinkgeschwindigkeit des Phytoplanktons $[\text{m}/\text{d}]$

XSEDI: Phytoplanktonverluste durch Sedimentation $[\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}]$

XSEDRA: Sedimentationsrate des Phytoplanktons $[\text{d}^{-1}]$

ZMIXD: Durchmischungstiefe $[\text{m}]$

Die Abhängigkeit der Sedimentationsgeschwindigkeit von der Größe und Form der Partikel sowie der Temperatur, der Dichte und der Viskosität des Wassers wird nicht berücksichtigt.

5.4.8. Sekundärproduktion

Im Submodell für die Sekundärproduktion werden von der Vielzahl der Glieder der Nahrungskette der Konsumenten die pelagischen Primärkonsumenten berücksichtigt, hierfür wiederum stellvertretend filtrierende (phytophage und omniphage) Cladoceren und Copepoden, die neben den Rotatorien die Hauptmenge des filtrierenden Zooplanktons im Pelagial ausmachen. Bei der Modellierung wird das bereits für das Grazing verwendete, sehr einfach gehaltene Submodell nach BENNDORF (1979) weitergeführt. Die Dynamik der Zooplanktonbiomasse bzw. -dichte Z ergibt sich aus der Differenz zwischen Wachstum (ZG) und Mortalität ($ZMORT$):

$$dZ = ZG - ZMORT \quad [cm^3/m^3]$$

Das Zooplanktonwachstum ZG errechnet sich aus der Wachstumsrate ZGR und der Zooplanktonbiomasse Z :

$$ZG = ZGR \cdot Z \quad [cm^3 \cdot m^{-3} \cdot d^{-1}]$$

Die Wachstumsrate ZGR ist die Differenz zwischen der Assimilationsrate $ZASSIM$ und der Respirationsrate $ZRESP$:

$$ZGR = ZASSIM - ZRESP \quad [d^{-1}]$$

Bei der Modellierung der Assimilationsrate ist vor allem die Frage von Bedeutung, in welchem Ausmaß die Produzenten für die Konsumenten greifbar und verwertbar sind und welcher Anteil der von den Konsumenten aufgenommenen Nahrung assimiliert werden kann. Zu berücksichtigen sind grundsätzlich die Filtrieraktivität der Zooplankter, die Freßbarkeit (Greifbarkeit und Ingestion) und Assimilierbarkeit der Algen sowie die Algenkonzentration. Im Modell hängt die Assimilationsrate $ZASSIM$ von der Verwertbarkeit des Futters und von der Fraßintensität $ZGZRA$ ab; mit steigender Grazing-Rate fällt die spezifische Futterausnutzung. Die Fraßintensität $ZGZRA$ wurde bereits erläutert.

$$ZASSIM = ZAK \cdot ZGZRA \quad [d^{-1}]$$

$$ZAK = \left[ZAKMAX - \left(\frac{ZAKMAX - ZAKMIN}{ZGZMAX} \right) \cdot ZGZRA \right] \cdot UXZ \quad [cm^3/cm^3]$$

ZAK: Verhältnis der assimilierten zur gefressenen Nahrungsmenge des Zooplanktons [cm^3/cm^3]

ZAKMAX: maximaler Assimilationskoeffizient bei sehr kleiner Grazing-Rate [cm^3/cm^3]

ZAKMIN: minimaler Assimilationskoeffizient bei $ZGZRA = ZGZMAX$ [cm^3/cm^3]

ZGZMAX: maximale Grazing-Rate [d^{-1}]

UXZ: Physiologische Verwertbarkeit des Phytoplanktons durch das Zooplankton [dimensionslos]

Die Respirationsrate ZRESP steigt von einem Grundumsatz mit zunehmender Grazing-Rate linear an und hängt exponentiell von der Wassertemperatur ab:

$$ZRESP = \left[\left(\frac{ZRESPOPT - ZRESPMIN}{ZGZMAX} \right) \cdot ZGZRA + ZRESPMIN \right] \cdot \frac{1}{ZRESPOPT} \\ \cdot \left[(ZRESPOPT - ZRSPTMIN) \cdot \left(\frac{TEMP^2}{ZGZTOPT} \right) + ZRSPTMIN \right] \quad [d^{-1}]$$

TEMP: Wassertemperatur ($^{\circ}C$)

ZGZMAX: maximale Grazing-Rate [d^{-1}]

ZGZRA: Grazing-Rate des Zooplanktons [d^{-1}]

ZGZTOPT: Optimaltemperatur für Grazing [$^{\circ}C$]

ZRESPMIN: Respirationsrate des Zooplanktons bei Optimaltemperatur und fehlendem Nahrungsangebot [d^{-1}]

ZRESPOPT: Respirationsrate des Zooplanktons bei Optimaltemperatur und maximaler Grazing-Rate [d^{-1}]

ZRSPTMIN: Respirationsrate des Zooplanktons bei Temperatur nahe $0^{\circ}C$ und optimalem Nahrungsangebot [d^{-1}]

Die Zooplanktonmortalität ZMORT errechnet sich aus der Mortalitätsrate ZMORA und der Zooplanktonbiomasse Z:

$$ZMORT = ZMORA \cdot Z \quad [cm^3 \cdot m^{-3} \cdot d^{-1}]$$

Die Mortalitätsrate des Zooplanktons ZMORA, die vor allem durch planktonfressende Fische bedingt ist, zeigt eine lineare Temperaturabhängigkeit sowie eine hyperbolische Zunahme mit steigender Zooplanktonbiomasse. Da die Fische

keine Systemvariable im Modell darstellen, korrespondiert die Mortalitätsrate für einen bestimmten Modellauf immer nur mit einer konstanten Fischpopulation. Die Fische sind aber insofern indirekt in das Modell einbezogen, als ihre Fraßleistung in die Mortalitätsrate (ZMORA) des Zooplanktons mit eingeht.

$$ZMORA = (ZMOMIN + ZMOT \cdot TEMP) \cdot \frac{Z}{KZMO + Z} \quad [d^{-1}]$$

KZMO: Halbsättigungskonstante für die Beziehung zwischen Zooplanktonbiomasse und Mortalitätsrate des Zooplanktons [cm^3/m^3]

TEMP: Wassertemperatur ($^{\circ}C$)

Z: Zooplanktonbiomasse (Dichte) [cm^3/m^3]

ZMOMIN: Zooplanktonmortalitätsrate nahe $0^{\circ}C$ bei großer Zooplanktonbiomasse ($Z \gg KZMO$) [d^{-1}]

ZMOT: Konstante der Temperaturabhängigkeit der Mortalitätsrate des Zooplanktons [$(^{\circ}C \cdot d)^{-1}$]

5.4.9. Sauerstoff

Für die Modellierung der im Wortmodell beschriebenen Dynamik der Konzentrationsänderungen des Sauerstoffs im Jahreslauf muß ein weiteres Submodell entworfen werden, das die vertikale Tiefenverteilung des Sauerstoffs pro Zeitschritt zeigen soll. Dabei tritt folgende Schwierigkeit auf: Bei dem im Modell *LAGO* entworfenen Submodell für die Primärproduktion erfolgt keine Quantifizierung des entstehenden Sauerstoffs. Daher können auch die in der wissenschaftlichen Systemanalyse häufig verwendeten Bilanzmodelle nicht verwendet werden, auch nicht in vereinfachter Form, da der wichtigste Posten in der Sauerstoffbilanz fehlt. Die Ankopplung eines Sauerstoff-Submodells an das aufgestellte Phytoplankton-Zooplankton-Phosphat-Grundmodell kann nur über die bereits vorhandenen Zustandsgrößen, Veränderungsraten oder Hilfsvariablen erfolgen. Da die für den Unterricht zumutbare Rechenzeit beim bisherigen Modellumfang bereits fast vollständig ausgeschöpft ist, kann die zeitliche Änderung der vertikalen Sauerstoffverteilung im Modellsee auch nicht durch eine weitere, in das bisherige Gleichungssystem integrierte Differentialgleichung ausgedrückt werden.

Als Lösung bietet sich an, im Sauerstoff-Submodell eine einfache Abschätzung der Sauerstoff-Zehrung vorzunehmen und mit dem Grundmodell über das die Zehrung verursachende sedimentierende Phyto- und Zooplankton zu verknüpfen: "Zur Abschätzung der Sauerstoffzehrung geht man von dem Sauerstoff-Vorrat aus, der sich während der Vollzirkulation in einem See gebildet hat. Die Zehrung drückt sich dann durch entsprechend hohen Verlust, die biogene Produktion durch einen Sauerstoff-Gewinn aus. Da Photosynthese und Zehrung gemeinsam in der trophogenen Zone stattfinden, in der tropholytischen Zone aber allein Sauerstoff-Zehrung erfolgt, sind in einem See in vertikaler Richtung erhebliche Sauerstoff-Konzentrationsunterschiede zu erwarten. Dieses antagonistische Geschehen verändert die zunächst physikalisch bedingte Sauerstoff-Verteilung somit mehr oder weniger stark." (SIEBECK 1982, S. 65).

Bei der Ermittlung der Sauerstoff-Zehrung muß noch die Entscheidung getroffen werden, welcher der drei möglichen Darstellungen des Sauerstoff-Defizits der Vorzug gegeben werden soll (vgl. WETZEL 1985, S. 138):

Das *absolute Sauerstoffdefizit* ist definiert als Unterschied zwischen der momentan beobachteten Sauerstoffkonzentration und dem Sauerstoffsättigungswert bei einer Wassertemperatur von 4 °C (und Normal-Luftdruck an der Wasseroberfläche). Es setzt die Annahme voraus, daß das Wasser in der Tiefe während der Frühjahrsvollzirkulation zu 100% mit Sauerstoff gesättigt wird und daß das Eindringen von Sauerstoff aus der trophogenen Zone während der Stratifikation vernachlässigt werden kann.

Das *relative Sauerstoffdefizit* ist definiert als Unterschied zwischen dem aktuellen Sauerstoffgehalt des Hypolimnions und dem Sauerstoffgehalt am Ende der Frühjahrsvollzirkulation oder, anders ausgedrückt, als Abweichung der Sauerstoffkonzentration von der Ausgangskonzentration vor Beginn der sommerlichen Schichtung.

Das *aktuelle Sauerstoffdefizit* ist definiert als Differenz zwischen dem beobachteten Sauerstoffgehalt zum momentanen Zeitpunkt und dem physikalischen Sättigungswert, der sich bei der entsprechenden Wassertemperatur einstellen würde. Es gelten die gleichen Einschränkungen wie beim absoluten Sauerstoffdefizit.

Für das Modell LAGO fällt die Entscheidung zugunsten des aktuellen Sauerstoffdefizits. Es läßt sich bei jedem Zeitschritt grafisch leicht verständlich veranschaulichen (vgl. Farbfoto, Anlage 9, S. 264): Die hellblaue Kurve im Koordinatensystem "Sauerstoffkonzentration" stellt die Tiefenverteilung des von der Wassertemperatur abhängigen maximalen Sättigungswertes dar und besitzt eine eigene, aussagekräftige Dynamik: Der Anwender kann über das gesamte Modelljahr die temperaturbedingten Veränderungen im Kurvenverlauf verfolgen und damit an einem typischen Beispiel die Auswirkungen physikalischer Gegebenheiten auf das Ökosystem "See" erkennen. Die rosafarbige Kurve zeigt die aktuelle, tiefenverteilte Sauerstoffkonzentration. Das aktuelle Sauerstoffdefizit wird als violette Fläche, eine Übersättigung als grüne Fläche gekennzeichnet. Ist die Sauerstoffkonzentration gleich Null, so wird die betroffene Tiefenschicht durch eine hellrote Schraffierung hervorgehoben. Die für die Verwendung des aktuellen Sauerstoffdefizits genannten einschränkenden Annahmen sind im Modell LAGO berücksichtigt.

Eine Entscheidung für das absolute Sauerstoffdefizit enthielte den Nachteil, daß sich das Defizit immer nur auf einen Festwert bezieht und damit die oben beschriebene Dynamik der wassertemperaturabhängigen Sättigungskurve wegfielen. Bei der Darstellung des relativen Sauerstoffdefizits ergäbe sich die Schwierigkeit, daß der Verlauf der Sauerstoffkonzentration zu Beginn der Sommerstagnation "eingefroren" werden müßte, was bei der Programmierung einen erheblichen zusätzlichen Aufwand verursachen würde.

Die Tiefenverteilung des von der Wassertemperatur abhängigen maximalen Sättigungswertes wird im Submodell "Sauerstoff" anhand einer Tabelle ermittelt, (vgl. Anlage 3, S. 146). Bei der Tiefe des Modellsees von 25 m kann auf eine Korrektur der Werte in Bezug auf den hydrostatischen Druck verzichtet werden.

Die aktuelle, tiefenverteilte Sauerstoffkonzentration wird mit Hilfe der sedimentierten Algen XSEDI und des abgestorbenen Zooplanktons ZMORT ermittelt. Die Summe aus diesen Biomassen wird mit Hilfe der Variablen CUMSED pro Dekade kumuliert, wobei berücksichtigt wird, daß sich die sedimentierte Biomasse durch Abbau verringert:

$$\text{CUMSED} = \text{CUMSED} + \text{XSEDI}/3 + \text{ZMORT}/3$$

Bei der Frühjahrs- und Herbstvollzirkulation wird die sedimentierte Biomasse zusätzlich verringert:

$$\begin{aligned} t = \text{FVZ} : \quad \text{CUMSED} &= \text{CUMSED} \cdot 0,7 \\ t = \text{HVZ} : \quad \text{CUMSED} &= \text{CUMSED} \cdot 0,85 \end{aligned}$$

t: Zeit

FVZ: Startdekade für Frühjahrvollzirkulation

HVZ: Startdekade für Herbstvollzirkulation

Die Menge an sedimentierter Biomasse CUMSED bestimmt das jeweils vorhandene Sauerstoffdefizit. Aus einer Tabelle, die in Anlehnung an Sauerstoff-Isoplethen verschiedener realer Seen gewonnen wurde (vgl. Anlage 3, S. 147), läßt sich dann die vertikale Tiefenverteilung der aktuellen Sauerstoffkonzentration in Abhängigkeit von CUMSED ermitteln und am Bildschirm grafisch veranschaulichen. Befindet sich der Modellsee im oligotrophen Zustand, nimmt sie

während der Stagnationsphase einen Verlauf an, der einer orthograden Kurve angenähert ist, ihr aber nicht voll entspricht, da man einen idealisierten orthograden Verlauf nur bei extrem unproduktiven Seen findet. Im Normalfall tritt während der Stagnation bei nicht zu tiefen, oligotrophen Seen aufgrund der Oxidationsprozesse ein Sauerstoffdefizit im Hypolimnion auf, das aber meist unter 50 % liegt (vgl. SCHWOERBEL 1984, S. 73). Geht der Modellsee in den eutrophen Zustand über, folgt die vertikale Tiefenverteilung der aktuellen Sauerstoffkonzentration während der Sommerstagnation mehr und mehr einer klinograden Kurve.

In Anlage 3 (S. 137 - 150) ist das vollständige mathematische Modell einschließlich der im Computerprogramm verwendeten Booleschen Funktionen und Tabellen enthalten.

5.5. Erstellung des Computerprogramms

5.5.1. Auswahl der Programmiersprache

Das Computerprogramm wird in der Programmiersprache PASCAL entwickelt. Als Compiler wird der Turbo-Pascal-Compiler der Firma Borland International, Version 3.0, verwendet. Für diese Entscheidung gibt es mehrere Gründe. Ich bin selbst im Besitz dieses Compilers einschließlich aller im Abschnitt "Programmierhilfen" (S. 33) genannter Tools. Außerdem verfüge ich über ausreichende Kenntnisse in der Programmiersprache PASCAL. Neben einem sehr günstigen Preis-Leistungs-Verhältnis hat der Turbo-Pascal-Compiler für die Entwicklung des Computerprogramms weitere Qualitäten, von denen ich einige kurz ansprechen möchte:

Turbo-Pascal entspricht bis auf wenige Ausnahmen voll der Standard-Definition aus dem "Pascal User Manual and Report" von Kathleen JENSEN und Nikolaus WIRTH. Neben einer schnellen und komfortablen Entwicklungsumgebung (z.B. integrierter Full-Screen-Editor, Kompilierung wahlweise direkt in den Speicher oder in eine COM- oder CHN-Datei, Fehlermeldung im Klartext mit Fehleranzeige im Programmcode) sind zusätzlich einige Spracherweiterungen enthalten, die die Programmierung des Computermodells LAGO wesentlich erleichtern. Hier sind vor allem die volle Unterstützung des Betriebssystems, die Erzeugung von Inline-Maschinencode, Assembler-nahe Prozeduren sowie die Möglichkeit, längere Programme mit Hilfe von Include-Dateien in Module zerlegen zu können, zu nennen. Turbo-Pascal-Programme laufen etwa mit gleicher Geschwindigkeit wie kompilierte BASIC-Programme, arbeiten aber mit einer höheren Rechengenauigkeit (elf gültige Stellen bei reellen Zahlen). Der Compiler ist für die Betriebssysteme MS-DOS, CP/M-80 und CP/M-86 lieferbar. Außerdem ist eine schnelle Anpassung an unterschiedliche Bildschirme möglich. Wenn beim Programmieren gerätespezifische Turbo-Pascal-Sprachelemente vermieden werden, sind die Programme daher ohne allzu zeitraubende Umarbeitung auf vielen unterschiedlichen Rechnern lauffähig.

Der Turbo-Pascal-Compiler hat allerdings im Vergleich mit anderen Compilern auch Nachteile: Er arbeitet nur als 1-Pass-Compiler und besitzt daher keinen echten Linker. Nach Änderungen am Programmcode muß jedesmal das gesamte Programm neu kompiliert werden. Bei einem umfangreichen Programm wird da-

durch viel wertvolle Programmierzeit vergeudet. Schwierigkeiten bereitet auch die Begrenzung des Code- und Datensegments auf jeweils 64 KByte. Diese Nachteile lassen sich aber durch die Nutzung spezieller Programmierverfahren und Hilfsprogramme ausgleichen. Lange Kompilierungszeiten können z.B. durch eine MODULA-ähnliche Zerlegung des Programmcodes in einzelne, getrennt kompilierbare Module vermieden werden. Um die Begrenzung des Code- und Datensegments zu überwinden, stehen drei verschiedene Möglichkeiten zur Verfügung:

1. Im Turbo-Pascal-Compiler ist ein OVERLAY-System enthalten, mit dem Programmteile, die im vorgesehenen Teil des Arbeitsspeichers keinen Platz finden, auf den Massenspeicher ausgelagert werden können. Dieses Verfahren bietet zudem den Vorteil, daß umfangreiche Programme auch auf Rechnern mit kleinem Arbeitsspeicher lauffähig sind, da die Gesamtgröße der ausgelagerten Programmteile die Kapazität des Hauptspeichers überschreiten kann. Allerdings muß bei der OVERLAY-Technik eine Laufzeiterhöhung, die durch die Häufigkeit der Zugriffe auf den Massenspeicher bestimmt ist, in Kauf genommen werden.

2. Turbo-Pascal erlaubt die Verwendung dynamischer Variablen und Zeiger. Während es die statischen Variablen in einem Daten-Segment, das die oben genannte Begrenzung aufweist, anlegt, werden dynamische Variable auf dem sogenannten *Heap*, das ist ein stapelartig strukturierter Abschnitt des Arbeitsspeichers, gehalten. Dieses Verfahren ist vor allem dann von Vorteil, wenn bei den Computern, auf denen das Programm eingesetzt werden soll, mit einer größeren Arbeitsspeicherkapazität (z.B. mehr als 256 KByte) gerechnet werden kann, da diese Kapazität in der Regel nicht voll genutzt wird. Allerdings muß beim Programmieren die Größe des *Heap* im Auge behalten werden, da er beim Speichern dynamischer Variablen aufwärts gegen den *Stack*, einen anderen Abschnitt des Arbeitsspeichers, wächst und daher eine Kollision auftreten kann, die zum Programmabsturz führt.

3. Hilfsprogramme, die einen Linker enthalten, erlauben ebenfalls die Ausnutzung des gesamten zur Verfügung stehenden Arbeitsspeichers. Sie unterteilen in der Regel den Programmcode, unter Berücksichtigung der Segmentarchitektur des Mikroprozessors, in zahlreiche Segmente. Der gesamte Code teilt sich dasselbe Datensegment und den *Heap*, so daß globale Variable und *Heap-variable* als Kommunikationsmittel zwischen Routinen in verschiedenen Code-

segmenten dienen können. Mit der Aufteilung in Codesegmente ist auch meist der Vorteil verbunden, daß sich die einzelnen Module getrennt kompilieren lassen.

Die Auswahl der Programmiersprache ist für die Programmentwicklung aber nicht kritisch. Akzeptable Lösungen lassen sich mit jeder anderen Sprache, sofern ein Compiler zur Verfügung steht, erreichen. In Frage kommen z.B. auch die neu auf den Markt gekommenen preisgünstigen Basic-, Modula- und C-Compiler, die aber bei Beginn der Programmentwicklung noch nicht erhältlich waren.

5.5.2. Erläuterungen zum Computerprogramm

Bei der Entwicklung des Computerprogramms wurden alle drei genannten Möglichkeiten, die Beschränkung des Code- und Datensegments zu umgehen, erprobt. Als vorteilhaft erwies sich vor allem die Verwendung der oben genannten Hilfsprogramme. Sie können tatsächlich, wenn häufige Änderungen am Programmcode vorgenommen werden müssen, enorm viel Kompilierungszeit einsparen. Als Grundversion des Computermodells *LAGO* soll jedoch ein mit *OVERLAY* - Dateien arbeitendes Programm gelten, da es am wenigsten Ansprüche an die Ausstattung der verwendeten Computer stellt. Diese Grundversion ist in Anlage 7 (S. 181 - 259) der Arbeit beigelegt. In den Programmcode sind, als Leihilfe, zahlreiche Kommentare eingefügt. Einige Besonderheiten des Programms sollen im folgenden kurz erläutert werden.

Wegen der genannten Begrenzung des Codesegments ist es nicht möglich, alle die im Vereinbarungsteil zu deklarierenden Prozeduren und Funktionen im Hauptprogramm *LAGO.PAS* (S. 182 - 205) unterzubringen. Viele von ihnen werden daher in *INCLUDE*-Dateien (*EGAPROC.INC*, *FONT4x6.INC*, *TSCHICHT.INC*) gepackt. Die Datei *EGAPROC.INC* (S. 241 - 247) enthält Routinen zur Ansteuerung des Grafikadapters. Hierzu folgen noch nähere Erläuterungen. Mit den Unterprogrammen der Datei *FONT4x6.INC* (S. 248 - 255) werden die in der Bildschirmgrafik erforderlichen kleinen Buchstaben und Zahlen (vgl. Farbfoto, Anlage 9, S. 246) aus einer Matrix von 4x6 Pixel generiert. Dieser etwas umständliche Weg ist notwendig, weil bereits zwei grafische Zeichensätze, der 8x8-ROM Zeichensatz und ein 9x14-Zeichensatz im Bildschirmspeicher vorhanden sind und damit die maximale Zahl von 512 Zeichen ausgeschöpft ist. Für die vorgesehene Bildschirmgrafik sind aber pro Bildschirmseite gleichzeitig alle drei Zeichensätze erforderlich. Die Datei *TSCHICHT.INC* (S. 256 - 259) enthält Prozeduren für den Auf- und Abbau der thermischen Schichtung sowie für die Ausgabe der Durchmischungspfeile bei Vollzirkulation.

Die Dateien *INTRO.OVL*, *TEMPVERT.OVL* und *HINWEISE.OVL* enthalten die als *OVERLAY* deklarierten Prozeduren. Da nur Programmteile, die beim Programmstart oder zu Beginn eines Modelljahres einmal benötigt werden, auf die Diskette ausgelagert werden, ist eine Erhöhung der Ausführungszeit des Programms durch die wenigen Zugriffe auf den Massenspeicher kaum meßbar. Die Prozeduren in der Datei *INTRO.OVL* (S. 206 - 232) dienen zur Erstellung des

Logos, zur Berechnung des von der veränderlichen Durchmischungstiefe abhängigen Volumens der durchmischten Schicht, zum Aufbau der Bildschirmgrafik sowie zur Ausgabe der diversen Auswahlmenüs. In der Datei TEMPVERT.OVL (S.233 - 237) sind die Werte für die Tiefenverteilung der Wassertemperatur und der maximalen Sauerstoffsättigung in zweidimensionalen Feldern abgelegt. Die Datei HINWEISE.OVL (S. 238 - 240) enthält schließlich die formatierten Texte, die während der Simulation, abhängig vom trophischen Zustand des Modellsees, am Bildschirm erscheinen.

Die für das Computermodell *LAGO* vorgesehene Bildschirmgrafik benötigt eine Mindestauflösung von 640 x 200 Bildpunkten (Pixel). Wenn der einfache Grafikadapter (bei IBM-kompatiblen Geräten als "CGA-Adapter" bezeichnet) verwendet wird, ist mit dieser Auflösung nur eine zweifarbige Darstellung möglich. Es wurde zwar eine vollständige Programmversion für diesen Adapter entwickelt, trotz des Einsatzes verschiedener Schraffierungen ist die Grafik aber sehr unübersichtlich. Wenn die für das Computermodell aufgestellte Forderung nach unterrichtstauglichen Visualisierungsmöglichkeiten erfüllt werden soll, muß eine verbesserte Lösung gesucht werden. Bei der vorgestellten Programmversion wird deswegen eine hochauflösende, mehrfarbige Grafik verwendet. Hierzu ist ein sogenannter "Enhanced Graphics Adapter" (EGA), der inzwischen zur Standardausstattung der neueren Rechner zählt, sowie ein dazu passender Bildschirm, erforderlich. Bei diesem Adapter stehen verschiedene Grafikmodi zur Wahl. Es können z.B. bei einer Auflösung von 640 x 350 Bildpunkten 16 Farben gleichzeitig dargestellt werden.

Im normalen Sprachumfang des Turbo-Pascal-Compilers sind derzeit noch keine Sprachelemente vorhanden, mit denen sich der Enhanced Graphics Adapter im erwünschten Bildschirmmodus ansprechen ließe. Die zur Ansteuerung des Adapters notwendigen Routinen müssen daher selbst erstellt werden. Diese Aufgabe ist nicht ganz einfach, da die EGA-Karte fast 70 Register aufweist und eine Vielfalt von Funktionen anbietet. Außerdem besteht immer die Gefahr, daß Programmierfehler zur Zerstörung des Adapters führen können.

Damit das Programm auf möglichst vielen der derzeit am Markt angebotenen EGA-Karten lauffähig ist, erfolgt die Ansteuerung über den Bildschirm-Interrupt 10H. Alle EGA-kompatiblen Grafikkarten verfügen mit ziemlicher Sicherheit über diesen Interrupt. Die anderen Möglichkeiten, den Adapter anzu-

steuern, z.B. das direkte Schreiben in den Speicherbereich oder die Verwendung von Routinen, die in ROM auf der Grafikkarte enthalten sind, können eventuell Kompatibilitätsprobleme verursachen: Wenn z.B. EGA-Karten verschiedener Hersteller unterschiedlich aufgebaut sind oder bei einer neueren Baureihe die Anfangsadressen der Routinen geändert werden, ist das Programm nicht mehr lauffähig und muß jeweils neu angepaßt werden, was in der Regel einen erheblichen Arbeitsaufwand erfordert. Die Interruptbehandlungs-Routine ist im Programm enthalten und somit vom Betriebssystem unabhängig. Die selbst entwickelten EGA-Routinen sind daher auch für Turbo-Pascal-Versionen unter dem Betriebssystemen CP/M-86 oder XENIX einsetzbar.

Da die Geschwindigkeit des Grafikaufbaus im Computermodell von erheblicher Bedeutung ist, werden häufig benützte, zeitkritische Routinen in ASSEMBLER programmiert und über die EXTERNAL-Anweisung in das Turbo-Pascal-Programm eingefügt. Hierbei ist vor allem zu beachten, daß erst bei der Übersetzung des Turbo-Pascal-Programmes entschieden wird, ab welcher Adresse der übersetzte Code gespeichert wird. Man muß also alle Anweisungen vermeiden, die auf absolute Adressen Bezug nehmen. Diese Einschränkung erschwert das Speichern von Daten. In einem Assembler-Programm wird normalerweise ein bestimmter, noch verschiebbarer Speicherbereich für die Daten reserviert. Der Linker sorgt dann dafür, daß die Befehle, die auf diese Daten zugreifen, der entgeltigen Lage der Daten angepaßt werden. Bei der Einbindung von Assembler-Routinen wird nur die übersetzte EGA-Assembler-Routine alleine gelinkt und das entstandene ausführbare Programm nach einer weiteren Umformung in den von Turbo-Pascal erzeugten Code eingefügt. Der Zugriff auf feste Speicherstellen zur Datenablage ist also unzulässig. Es bietet sich daher an, den Daten einen Bereich auf dem Stack zu reservieren. Man erniedrigt dazu den Stapelzeiger in der Weise, daß der entsprechende Bereich nicht überschrieben wird, wenn weitere Werte auf dem Stack abgelegt werden. Dann kopiert man den Stapelzeiger in ein Indexregister (SI oder DI) und kann alle Daten durch den bereits zum Zeitpunkt der Programmmentstehung bekannten Offset zum Indexregister ansprechen. Beim Rücksprung aus der Assembler-Routine ins Turbo-Pascal-Programm ist zu beachten, daß von der Assembler-Routine der Stapel von den übergebenen Parametern freigemacht werden muß. Dies erreicht man durch die Anweisung "RET n", wobei n der Zahl der durch die Parameter auf dem Stapel belegten Bytes entspricht. Alle zur Ansteuerung der EGA-Karte

entwickelten Prozeduren und Funktionen sind in der INCLUDE-Datei EGA-PROC.INC (S. 241 - 247) enthalten und kommentiert.

Nachdem sich gezeigt hat, daß die Berechnung der Werte der Sinuskurven, die bei der grafischen Darstellung des Seenbeckens verwendet werden, fast zwei Minuten benötigt, werden diese Werte in Form einer sequentiellen Datei (WLY.DAT) auf der Diskette abgespeichert. Mit den Werten für die Darstellung der Durchmischungspfeile wird genauso verfahren (Datei FLASH_VZ.DAT). Die Prozeduren "read_datei_waterline" (in Datei INTRO.OVL, S. 206) bzw. "array_flash_vz" (in Datei LAGO.PAS, S. 190) lesen die Werte in Arrays ein, die dann für die Bildschirmgrafik abrufbereit zur Verfügung stehen. Als weitere Maßnahme zur Laufzeitoptimierung werden die Werte in den Compilerbefehlen so eingestellt, daß eine möglichst hohe Ausführungsgeschwindigkeit erreicht wird. Laufzeitkritisch sind vor allem der U-Befehl, der die Programmunterbrechung durch den Benutzer kontrolliert, der C-Befehl, der die Interpretation der Kontrollzeichen während der Ein-/Ausgabe durch die Konsole steuert sowie der R-Befehl, der die Indexprüfung zur Laufzeit vornimmt. Die genannten Compilerbefehle sind in den Passivmodus gesetzt.

Wenn das Programm über die Menüs vom Anwender Eingaben anfordert, so werden diese Eingaben überprüft, ob der verlangte Datentyp vorhanden und der zulässige Wertebereich eingehalten worden ist. Unzulässige Werte werden mit einem akustischen Warnton zurückgewiesen (vgl. z.B. Prozedur "einwohner_neu" in Datei INTRO.OVL, S. 211). Fehlerhafte Eingaben können daher nicht zu einem Überschreiben des Menüs bzw. zu einer Bereichsüberschreitung und bei letzterer unter Umständen zu einem Programmabsturz führen. Im übrigen wird auf eine unmißverständliche, selbsterklärende, auch für den Ungeübten problemlose Bedienerführung geachtet: Zur Auswahl der Menüpunkte dienen Ziffern, bei der Eingabe von Werten erhält der Benutzer stets Hinweise über den zulässigen Wertebereich. Ein Anhalten des Programms ist mit der Taste "S" möglich, durch Drücken einer beliebigen Taste wird das Programm fortgesetzt. Ein versehentliches Beenden des Programms soll durch eine nochmalige Bestätigung dieser Absicht vermieden werden.

Die gesamte Grafik ist in Bereiche eingeteilt, die sich bei Bedarf leicht verändern lassen (vgl. Datei LAGO.PAS, Anweisungsteil, S. 203). Auch die bei der Darstellung verwendeten Farben sind mit Hilfe einer Farbpalette (Datei IN-

TRO.OVL, S. 218) schnell zu bestimmen. Damit läßt sich die Grafik ohne großen Aufwand an andere Monitore mit einer vom EGA-Standard abweichenden Zahl von horizontalen und vertikalen Pixels anpassen.

Häufig ist nur ein kleiner Teil eines Programms für die Gesamtlaufzeit entscheidend. Es ist aber nicht immer ganz einfach, die laufzeitkritischen Stellen in einem umfangreichen Programmcode festzustellen. Um zu vermeiden, daß eine Menge Arbeit in die Optimierung unkritischer Bereiche investiert wird, habe ich für die Laufzeitanalyse des vorliegenden Programms ein Hilfsprogramm verwendet, das die in jeder Prozedur und Funktion verbrachte Zeit mit einer Genauigkeit von 200 Mikrosekunden ermittelt. Wird eine MS-DOS-Funktion aufgerufen, wird die benötigte Zeit der rufenden Routine zugeschrieben. Es läßt sich daher auch feststellen, welche Zeit Bildschirm- und Diskettenzugriffe benötigen. Auf diese Weise können alle laufzeitkritischen Programmteile erkannt und gezielt überarbeitet werden.

In Anlage 8, S. 260 - 263 ist das Protokoll der Laufzeitanalyse des beiliegenden Computerprogramms enthalten. Das Programm ist bereits so gut wie möglich laufzeitoptimiert. Spalte 1 zeigt die prozentuale Verteilung der Laufzeit. Hierbei muß berücksichtigt werden, daß das für die Laufzeitanalyse modifizierte Programm um den Faktor 2.7 langsamer läuft. Das prozentuale Verhältnis der für die aufgelisteten Prozeduren und Funktionen benötigten Zeit bleibt davon aber unberührt. In der zweiten Spalte ist die Anzahl der Aufrufe für jede Prozedur oder Funktion genannt, in der Spalte 3 die jeweilige Rekursionstiefe angegeben. Letztere ist für die Optimierung von Geschwindigkeit und Speicherplatz von Interesse. Die vierte Spalte gibt die gesamte, während des Programmlaufs in der Prozedur oder Funktion verbrachte Zeit an, in der fünften Spalte ist die Durchschnittszeit für einen Aufruf aufgeführt. Die letzte Spalte enthält den Namen der Prozedur oder Funktion.

Die Laufzeitanalyse zeigt, daß ein großer Teil der Zeit für die Prozeduren "model", sowie "licht_und_temperatur" benötigt wird (45%). Diese Unterprogramme enthalten zusammen mit der aufrufenden Prozedur "runge_kutta" den Kern des mathematischen Modells, d.h. hier erfolgt die numerische Approximation des verwendeten Differentialgleichungssystems. Dieser Programmteil war für die Laufzeitoptimierung entscheidend. Durch die in Abschnitt 5.4.1., S.52 erläuterte Einführung einer variablen Schrittweite konnte die Laufzeit um den

Faktor 10 verbessert werden. Mit ca. 18% schlägt ein Teil der Bildschirmgrafik, die Ausgabe der Kurven für die Sauerstoffkonzentration und die farbliche Kennzeichnung des aktuellen Sauerstoffdefizits zu Buche. Auch hier waren die Laufzeitwerte zunächst um den Faktor 2 höher. Eine Optimierung konnte durch die Programmierung häufig benutzter Routinen für die Ausgabe der Bildschirmgrafik in Assembler sowie durch die Verwendung schnellerer Algorithmen für das Zeichnen der Kurven erreicht werden.

Auch das nach dem Kompilieren direkt ausführbare .COM-File kann einer Laufzeitoptimierung unterzogen werden. Hierzu habe ich ein weiteres Hilfsprogramm, einen "Optimizer" verwendet. Er entfernt zum einen unnütze Anweisungen wie z.B. Sprünge der Länge Null und unnötige Register-Ladeanweisungen, die selbst bei den besten Compilern auftreten. Die Funktion des Optimizers ist mit der von Compilern hochentwickelter Mainframes und Minicomputer vergleichbar, wo ein Post-Prozessor (Peep-Hole-Optimizer) diese Arbeit erledigt. Das Hilfsprogramm scannt weiterhin den Programmcode der Runtime-Library durch und entfernt die nicht benötigten Teile dieses Laufzeitsystems. Diese Optimierungsmaßnahmen führten zu einer weiteren Laufzeitverbesserung von ca. 8%.

5.6. Darstellung des Computermodells

Nach den bisherigen Phasen der Modellentwicklung liegt nun ein funktionsfähiges und optimiertes Computerprogramm vor. Bevor als letzter Bearbeitungsschritt die Evaluation erfolgt, soll in diesem Abschnitt erläutert werden, auf welche Weise der Benutzer mit dem Computermodell *LAGO* arbeiten kann und wie sich ihm Struktur und Verhalten des Modells präsentieren. Hierzu wird folgender Weg der Dokumentation gewählt:

Anlage 6 (S. 161 – 180) enthält zahlreiche mehrfarbige Ausdrücke (Hardcopies), die mit der Grafik auf dem Bildschirm fast identisch sind. Abweichungen betreffen nur die Anzahl der Farben ³¹⁾. Einen originalen Eindruck von der Bildschirmgrafik vermittelt das Farbfoto in Anlage 9, S. 264. Anhand der Ausdrücke soll das Verhalten des Computermodells bei unterschiedlichen Parametereinstellungen aufgezeigt werden. Alle zeitlichen Zustände lassen sich auf diese Weise nicht dokumentieren; ihre Anzahl (pro Modelljahr 36) würde den Umfang der Arbeit sprengen. Da die Ausdrücke auch als Grundlage für die Validation (vgl. Abschnitt 5.7.1., S. 110) des Computermodells dienen, wurden solche Zustände gewählt, die hierfür besonders aussagekräftig sind. Die Menüs, über die der Benutzer die verschiedenen Parameter verändern kann, sind in Anlage 5 (S. 153 – 160) enthalten.

Zunächst wird ein Modelljahr detailliert dargestellt, da es für Vergleiche mit Modellläufen unter veränderten Parameterwerten als Referenz dienen soll. Für das Referenzjahr gelten folgende Einstellungen:

- NORMALLAUF (vgl. Menü 1, S. 154)
- Ausgangszustand des Sees: NÄHRSTOFFARM (vgl. Menü 3, S. 156)
- klimatische Bedingungen: NORMALER SOMMER, NORMALER WINTER, (vgl. Menü 4, S. 157)
- Phosphatimport 43,4 mg/m³ (vgl. Menü 5, S. 158); dieser Wert errechnet sich aus den im Menü voreingestellten Größen für Einwohnerzahl, gedüngte Wiesen und Ackerflächen, Art der Abwasserbehandlung usw.

Der Modellsee befindetet sich also zu Beginn der Simulation im oligotrophen Zustand. Nach 20 Tagen (Ausdruck 1, S. 163) hat sich die Oberfläche des Sees der Jahreszeit entsprechend abgekühlt. Es beginnt sich

³¹⁾ Von den sechzehn in der Bildschirmgrafik enthaltenen Farben kann der verwendete Drucker nur acht darstellen.

eine inverse thermische Schichtung auszubilden. Durch den Phosphatimport im Zufluß von $43,4 \text{ mg/m}^3$ hat sich die Orthophosphatkonzentration im Wasserkörper von $1,0 \text{ mg/m}^3$ (Anfangswert im oligotropen Zustand) bereits auf $2,3 \text{ mg/m}^3$ erhöht. Da der Abfluß des Modellsees immer aus der durchmischten Schicht erfolgt, haben Konzentration im Wasserkörper und Konzentration im Abfluß stets den gleichen Wert. Die Abbauprozesse führen zu einem leichten aktuellen Sauerstoffdefizit (violett gekennzeichnete Fläche).

Während der ersten Dekade des Monats Februar (Ausdruck 2, S. 164) ist der Modellsee noch von einer Eis- und Schneedecke bedeckt. (Sie ist auf dem Bildschirm natürlich weiß gekennzeichnet.) Die inverse Schichtung hat sich voll ausgebildet. Obwohl die Intensität der Strahlung über der Wasseroberfläche langsam zunimmt, führt die Eis- und Schneedecke zu einer Minderung der Strahlungsintensität im Wasserkörper (vgl. rote Balken im Histogramm "Licht").

Ausdruck 3 (S. 165) zeigt den Modellsee im Zustand der Frühjahrsvollzirkulation. Der Wasserkörper hat in allen Tiefen die gleiche Temperatur. Sauerstoff wird bis in die tieferen Schichten des Wasserkörpers getragen und gleicht das vormals vorhandene schwache Sauerstoffdefizit aus.

Im Monat Juni (Ausdruck 4, S. 166) haben sich infolge der steigenden Wassertemperatur und der erhöhten Lichtintensität die Algen zu einem Frühjahrsmaximum entwickelt. Es ist auch bereits ein Ansteigen der Zooplanktonbiomasse zu beobachten. Die Algen haben das gelöste Orthophosphat in der durchmischten Schicht deutlich reduziert. In der trophogenen Zone hat die Zunahme des Phytoplanktons zu einer Verminderung der Lichtintensität infolge Selbstbeschattung und zu einer Sauerstoffübersättigung (grün gekennzeichnete Fläche) geführt. Die steigenden Wassertemperaturen verursachen außerdem die Ausbildung der thermischen Schichtung; es sind bereits Epilimnion (hellblau), Metalimnion (violett) und Hypolimnion (dunkelblau) zu erkennen. Die Schichtung hat sich, was wegen der fehlenden Darstellung der dazwischenliegenden Zeitschritte nicht erkennbar ist, von Dekade zu Dekade langsam aufgebaut.

Anfang August (Ausdruck 5, S. 167) ist bei den gewählten klimatischen Bedingungen mit 22°C die maximale Temperatur an der Wasseroberfläche und damit der Höhepunkt der Sommerstagnation erreicht. Die sedimentierenden Biomassen von Phyto- und Zooplankton führen zu einem, im Vergleich zum Jahresanfang stärkeren aktuellen Sauerstoffdefizit.

Die Herbstvollzirkulation setzt Anfang November ein und dauert bis Anfang Dezember (Ausdruck 6, S. 168). Die Orthophosphatkonzentration des Wasserkörpers hat sich bei der Durchmischung erhöht. Der in die Tiefe getragene Sauerstoff gleicht das aktuelle Sauerstoffdefizit wieder aus. Der Zustand Ende Dezember (Ausdruck 7, S. 169) gibt schließlich einen Gesamtüberblick über den Jahresverlauf der Systemkomponenten Licht, Orthophosphatkonzentration, Phytoplankton und Zooplankton.

Bei den folgenden Modellläufen wurde gegenüber dem Referenzjahr nur jeweils ein Parameter verändert.

Aus dem Jahresüberblick Ende Dezember in Ausdruck 8 (S. 170) ist zu ersehen, daß sich bei Wahl der klimatischen Bedingungen "warmer Sommer/warmer Winter/geringe Bewölkung" eine Sinuskurve mit deutlich erhöhten Werten für die Lichtintensität über und unter der Wasseroberfläche ergibt. Der Frühjahrsgipfel der Algen bildet sich eine Dekade früher aus. Die gesamte Biomasse von Phyto- und Zooplankton ist, über das ganze Jahr gerechnet, leicht erhöht. Bei den klimatischen Bedingungen "kühler Sommer / kalter Winter / häufige Bewölkung" ist die Lichtintensität gegenüber dem Referenzjahr niedriger und das vermehrte Algenwachstum setzt ca. zwei Dekaden später ein (Ausdruck 9, S. 171).

Die nächsten beiden Ausdrücke zeigen ein Modelljahr, bei dem der Phosphatimport auf 6,3 mg P/m³ eingestellt wurde. Im Monat Juni (Ausdruck 10, S. 172) tritt gegenüber dem Referenzjahr (Ausdruck 4, S. 166) infolge der deutlich geringeren Algenbiomasse keine Sauerstoffübersättigung in der trophogenen Zone auf; die geringe Orthophosphatkonzentration wirkt limitierend. Auch die Sauerstoffzehrung im Hypolimnion ist geringer, da sich, wie der Jahresüberblick (Ausdruck 11, S.

173) zeigt, über das ganze Modelljahr gesehen, wesentlich weniger Phyto- und Zooplankton entwickelt hat.

Ein längerer Simulationslauf, der sich vom Referenzjahr ausgehend über mehrere Modelljahre erstreckt, soll nun das Modellverhalten bei einer raschen anthropogen bedingten Eutrophierung zeigen.

Die Simulation wird zunächst mit den für das Referenzjahr gewählten Parametern (vgl. S. 105) weitergeführt. Ausdruck 12 (S. 174) zeigt den Modellsee im 3. Jahr, Monat Juni. Im Vergleich mit Ausdruck 4 (S. 166) sieht man die Auswirkungen erhöhten Nährstoffimports bereits deutlich. Die Sauerstoff-Übersättigung in der trophogenen Zone ist höher, ebenso das aktuelle Sauerstoffdefizit im tropholytischen Bereich. Die Zunahme der Biomasse von Phyto- und Zooplankton hat eine stärkere Sedimentation zur Folge. Im Monat August (Ausdruck 13, S. 175) kommt es daher am Seegrund zu anaeroben Verhältnissen und damit zur Phosphatfreisetzung aus dem Sediment. Dies wird durch die PO_4 -Pfeile sowie durch einen Texthinweis verdeutlicht. Durch die "interne Düngung" ergibt sich für die Orthophosphatkonzentration bei der Vollzirkulation im Herbst ein höherer Wert als im Frühjahr (vgl. Ausdruck 14, S. 176).

Um nochmals die Sensitivität des Modells auf unterschiedliche klimatische Bedingungen zu zeigen, wird das 3. Modelljahr mit dem Parameter "warmer Sommer/warmer Winter/geringe Bewölkung" wiederholt. Die Auswirkungen sind in den Ausdrücken 15 und 16 (S. 177 und 178) zu sehen: Die Biomasse von Phyto- und Zooplankton ist höher, das aktuelle Sauerstoffdefizit größer.

Ab dem vierten Modelljahr wird der Phosphatimport auf $75,8 \text{ mg/m}^3$ erhöht. Der Modellsee gerät in den hypertrophen Zustand: Algen und Zooplankton nehmen rapide zu. Die anaerobe Zone breitet sich immer mehr nach oben aus (vgl. 5. Modelljahr, Ausdruck 17, S. 179). Auf dem Bildschirm erscheint ein Hinweis auf die rapide Verschlechterung der Wasserqualität infolge der Entstehung von Schwefelwasserstoff. Die starke Nährstoffbelastung führt schließlich im 6. Modelljahr zum "Umkippen" (Ausdruck 18, S. 180) des Sees: Beim Einsetzen der Vollzirku-

lation im Herbst ist auch in der trophogenen Zone die Konzentration des Schwefelwasserstoffs so hoch und die Sauerstoffkonzentration so gering (ca. 2 mg/l), daß die Lebensbedingungen für die höheren Organismen nicht mehr gegeben sind. Die Simulation wird mit einem Alarmton und einem Hinweis auf diesen Zustand abgebrochen.

5.7. Evaluation des Computermodells

5.7.1. Überprüfung der Modellgüte

Zur Überprüfung der Modellgüte muß das Computermodell verifiziert und validiert werden. Auf den Seiten 30 und 31 wurde die für didaktische Zwecke modifizierte Anwendung dieser Testverfahren bereits erläutert. Die dort genannte Schwierigkeit bei der Verifikation eines didaktischen Modells gilt auch für das Computermodell *LAGO*: Es soll keinen bestimmten, real existierenden See repräsentieren, sondern allgemeine Aussagen über den dimiktischen, holomiktischen, durch eine relativ lange Wassernerneuerungszeit gekennzeichneten Seentyp machen. Für den Vergleich von Modell und Realität sind daher idealisierte, aus einem größeren Spektrum solcher Seen gewonnene Werte erforderlich.

Für die Verifikation des Computermodells werden diese idealisierten Werte aus verschiedenen Veröffentlichungen, die entweder Meßergebnisse über Untersuchungen an verschiedenen Seen oder bereits verifizierte und validierte Ergebnisse von Modellen oder Teilmodellen enthalten, gewonnen ³²). Bei der Entwicklung des mathematischen Modells ist bereits an einigen Stellen auf weitere Quellen hingewiesen worden (vgl. S. 65, 68, 73, 76, 80).

Für die Validation ist ein eingehender Vergleich von tatsächlichem Modellverhalten mit der entsprechenden Beschreibung im Wortmodell erforderlich. Hierfür sind einige hundert Testläufe durchgeführt worden. Die zur Dokumentation der Validität besonders geeigneten "Momentaufnahmen" während verschiedener Simulationsläufe wurden mit den in Abschnitt 5.6. kommentierten 18 Ausdrucken (S. 161 ff.) bereits vorgestellt. Als Ergebnis des Vergleichs kann festgehalten werden: Das Modellverhalten entspricht in vollem Umfang der Beschreibung im Wortmodell. Durch die hohe Zahl von Rückkopplungen kann das Computermodell die wechselseitigen Abhängigkeiten der ausgewählten Systemfaktoren Licht (über und unter der Wasseroberfläche), Wassertemperatur, Orthophosphatkonzentration, Phytoplankton, Zooplankton und Sauerstoffkonzentration (Übersättigung und aktuelles Defizit) wie vorgesehen wiedergeben. Die Validität gilt auch für die Absicht, die typischen Folgen einer durch hohen Nährstoffimport

32) Verwendet werden z.B. Untersuchungen über den Starnberger See (Bayerisches Landesamt für Wasserwirtschaft 3/82), den Königsee (Der Königsee 1982), den Fischkälter-, Weflinger- und Obinger See (Bayerisches Landesamt für Wasserwirtschaft 1/82) sowie Meßergebnisse, die über mehrere Jahre mit Studentengruppen unter der Leitung von Prof. Dr. SIEBCK am Seener Klostersee erarbeitet worden sind.

Zahlreiche Werte sind an den Daten im "Handbook of environmental data and ecological parameters" (JØRGENSEN u.a., 1979) orientiert.

bedingten Eutrophierung aufzuzeigen. Weiterhin weist das Computermodell eine ausreichende Sensitivität bezüglich der vom Benutzer wählbaren Parameter auf. Die Auswirkungen der unterschiedlichen klimatischen Bedingungen auf das Modellverhalten wurde mit den genannten Ausdrücken bereits gezeigt. Mit Hilfe vieler Testläufe konnte auch die Validität für den Fall bestätigt werden, daß mit dem Modell experimentiert werden soll (vgl. Menü 2, S. 165).

Zur weiteren Überprüfung und Optimierung der Modellgüte konnte das Computermodell im Juli 1986 während eines limnologischen Praktikums (Leitung Prof. Dr. Siebeck), einer Veranstaltung des Zoologischen Instituts der Universität München, vorgestellt und diskutiert werden. Teilnehmer dieses Kurses waren 22 Diplomstudenten (Biologie). Die Diskussion ergab noch einige wertvolle Gesichtspunkte für die Validierung des Computermodells. Die Teilnehmer waren der Meinung, daß das Modellverhalten in hohem Maß als allgemeingültig gelten kann. Als fehlerhaft wurde dagegen der Zeitpunkt, bei dem der Modellsee umklippen sollte, bezeichnet. Anhand von Beobachtungen und Messungen, die in der Limnologischen Station der Universität München in Seeon an verschiedenen Seen durchgeführt worden sind, konnte das Modell in diesem Punkt inzwischen korrigiert werden. An weiteren Verbesserungsvorschlägen zum Modellverhalten wurden genannt: die Orthophosphatkonzentration könnte während der Stagnationsphase niedriger und der Herbstgipfel des Zooplanktons deutlicher ausgeprägt sein. Um diese beiden Vorschläge zu realisieren, wurde eine Optimierung der betreffenden Parameter versucht, eine Lösung, die das Modellverhalten nicht in anderen Punkten beeinträchtigt, aber (bisher) nicht gefunden.

5.7.2. Erprobung im Unterricht

Ziel und Methode

Ziel der unterrichtlichen Erprobung ist es, festzustellen, ob das ComputermodeLL die vorgesehenen didaktischen Funktionen und die in Kapitel 3 genannten Kriterien erfüllt. Im einzelnen sind folgende Fragen von besonderem Interesse:

1. Ist es gelungen, das Zusammenwirken biotischer und abiotischer Faktoren im Ökosystem "See" zu veranschaulichen ?
2. Bietet das Modell genügend Interaktionsmöglichkeiten für eine aktive Auseinandersetzung mit dem gegebenen Sachverhalt ?
3. Ist das ComputermodeLL hinsichtlich seines Schwierigkeitsgrades der Oberstufe des Gymnasiums (Sekundarstufe II) angemessen ?
4. Besitzt das ComputermodeLL ausreichenden Motivationscharakter?
5. Ist die gewählte Darstellung verständlich? War das Vorwissen bzw. waren die vor und während der Simulation gegebenen Zusatzinformationen ausreichend ?
6. Genügt das Modell den aufgestellten Anforderungen an die Bedienungs-freundlichkeit?

Da der Schwerpunkt der vorliegenden Arbeit auf der Entwicklung eines ComputermodeLLs liegt, ist eine Erprobung in Form einer umfangreichen empirischen Untersuchung nicht vorgesehen. Letztere soll in einer weiterführenden Arbeit durchgeführt werden. Für die hier beabsichtigte Evaluation des Modells wird eine *erste Einschätzung* von Seiten der Schüler und der Lehrer über die Eignung des ComputermodeLLs für den Biologieunterricht auf der vorgesehenen Schulstufe als ausreichend erachtet.

Durchführung der Erprobung

Die unterrichtliche Erprobung erfolgte im April 1987 in drei Leistungskursen der 13. Jahrgangsstufe am Karls gymnasium in München; dabei wurden insgesamt 42 Schülerinnen und Schüler erfaßt.

Bei einem der Kurse konnten die für die Thematik vorgesehenen Lernziele ³³⁾ in methodisch optimaler Weise erarbeitet werden: Die Schülerinnen und Schüler unternahmen im Juni 1986 eine ganztägige Exkursion zum Seeoner Klostersee (Nähe Chiemsee/Oberbayern). Die Gruppe untersuchte am Vormittag das Pelagial des Sees. Mit einem RUTTNER-Schöpfer wurden Wasserproben vertikal in Abständen von einem Meter entnommen, die Temperatur ermittelt und verschiedene chemische Analysen mit AQUAMERCK*-Testsätzen ³⁴⁾ durchgeführt (quantitative Bestimmung von Sauerstoff, Wasserstoffionenkonzentration, Ammonium, Nitrit, Nitrat, Calcium, Gesamthärte, Gesamtkohlenstoff). Die Teilnehmer entnahmen außerdem aus verschiedenen Tiefen Planktonproben und füllten Parallelproben zur Bestimmung des BSB₅ in WINKLERSche Meßflaschen ab. Am Nachmittag nahmen die Schülerinnen und Schüler im Labor der am Klostersee gelegenen Limnologischen Station der Universität München die mikroskopische Bearbeitung der Planktonproben und die Auswertung der Meßergebnisse vor. Im weiteren Verlauf der Unterrichtssequenz wurden in der Schule die übrigen, im Lehrplan geforderten Lerninhalte erarbeitet.

Die beiden anderen Leistungskurse haben den Unterrichtsstoff ohne Exkursion erarbeitet.

Im Frühjahr 1987 stand dann für die Leistungskurse die Wiederholung des Lernstoffes zur Vorbereitung auf die Reifeprüfung an. Bei der Wiederholungsarbeit zum Ökosystem "See" wurde das Computermodell *LAGO* eingesetzt und erprobt. Hierfür stand pro Kurs eine Unterrichtsdoppelstunde (90 Minuten) zur Verfügung. Nach einer kurzen Zusammenfassung dessen, was im Vorjahr erarbeitet worden ist (ca. 20 Minuten), wurde intensiv mit dem Computermodell gearbeitet. Mit einem hochwertigen Großbildprojektor konnte die Bildschirmgrafik mit der Qualität des normalerweise verwendeten Farbbildschirms (vgl.

³³⁾ Lernziele 5.2. und 5.3. im bayerischen curricularen Lehrplan für das Fach Biologie in der Kollegstufe des Gymnasiums. KIMBL I, So.-Nr. 17/1979, München, 1979, S. 527 und 528

³⁴⁾ * AQUAMERCK ist eingetragenes Warenzeichen der Firma E. Merck, Darmstadt

Anlage 9, S. 264) auf eine Fläche von 2 x 3 Meter projiziert und somit der gesamten Schülergruppe zugänglich gemacht werden. Die Erprobung erfolgte für jeden der drei Leistungskurse getrennt. Die Schülerinnen und Schüler konnten die Simulation jeweils über mindestens acht Modelljahre verfolgen und dabei verschiedene Parametereinstellungen vornehmen. Bewertungen über das Computermodell erfolgten in dieser Phase noch nicht. Es sei noch angemerkt, daß die Schüler bisher weder im Biologieunterricht noch in einem anderen Fach mit einem Simulationsprogramm gearbeitet hatten.

Anschließend wurden die Schülerinnen und Schüler gebeten, ihre spontanen Eindrücke über die Arbeit mit dem Computermodell auf einem anonymen Fragebogen festzuhalten (vgl. Anlage 4, S. 151/152). Die schriftliche Befragung wurde gewählt, damit die Schülerinnen und Schüler in der zur Verfügung stehenden kurzen Zeitspanne, unbeeinflusst von einer allgemeinen Diskussion, ihre spontanen, persönlichen Eindrücke niederlegen können. Die Fragen 3 - 5 des Fragebogens gehören zur Klasse der offenen Aufgaben, bei denen die Antworten nicht vorgegeben sind. Die Schülerinnen und Schüler sollten aus der breiten Palette denkbarer Möglichkeiten die ihnen wesentlich erscheinenden positiven oder negativen Charakteristika des Modells selbst erkennen. Eine spontane und doch qualifizierte Bewertung eines Unterrichtsmediums ist für die Schüler der Kollegstufe ohne weiteres leistbar. Im Anschluß an die schriftliche Befragung konnten in einer Aussprache weitere, auf dem Fragebogen nicht enthaltene Fragen und Bewertungsgesichtspunkte besprochen werden.

Ergebnisse

a) Auswertung der Fragebogen (Fragebogen siehe Anlage 4, S. 151/152):

Zahl der bearbeiteten Fragebogen: 42

Frage 1: War Ihr Vorwissen für die Arbeit mit dem Computer-Modell ausreichend?

ja: 42

nein: 0

Frage 2: Schwierigkeitsgrad?

überfordernd	0
schwer	1
gerade richtig ...	40
leicht	1
zu leicht	0

Frage 3: Welche Vorteile bzw. Nachteile bietet Ihrer Meinung nach die Arbeit mit dem Computermodell?

Antworten	Anzahl
a) Vorteile	
■ hohe Anschaulichkeit	20
■ Zusammenhänge sind leichter erkennbar	16
■ Durch die Veränderbarkeit verschiedener Parameter können unterschiedliche Ausgangssituationen durchgespielt werden	9
■ das Problembewußtsein wird gestärkt	4
■ Die Arbeit mit dem Modell weckt Interesse für den Sachverhalt	3
■ Die Simulation ist eine willkommene Abwechslung	3
■ Das Modell eignet sich besonders für die Zusammenfassung/Wiederholung/Vertiefung des Unterrichtsstoffes	2
b) Nachteile	
■ Das Modell ist nicht geeignet zur reinen Stoffvermittlung	1

Frage 4: Wurden Inhalte vermittelt, die andere Lehrmedien nicht vermitteln können?

nein: 11 ja: 29

Antworten	Anzahl
■ Das Verhalten eines Systems kann unter <i>verschiedenen Bedingungen</i> getestet werden	10
■ Das Modell zeigt <i>dynamische Zusammenhänge</i> , nicht nur starre Ausschnitte	9
■ Die Bedeutung <i>einzelner Faktoren</i> für den <i>Gesamtzusammenhang</i> wird deutlich; das Modell bietet einen Gesamt-Überblick	8
■ Ein Prozeß kann über <i>längere Zeit</i> verfolgt werden	5

■ Es werden <i>keine neuen Inhalte</i> vermittelt, sondern es wird nur die Verständlichkeit erhöht	1
■ Verweis auf Frage 3	6

Frage 5: Wurden durch das Computermodell Zusammenhänge im Ökosystem besonders deutlich?

nein: 1 ja: 41

Antworten	Anzahl
An Zusammenhängen wurden konkret genannt:	
■ die Rolle des Phosphats im Stoffkreislauf/ die Wirkung erhöhten Phosphatimports	11
■ der Einfluß anthropogen bedingter Faktoren auf ein Ökosystem	5
■ die Einflüsse der Strahlung auf die Algenbiomasse	4
■ die Abhängigkeit des Sauerstoffgehalts von verschiedenen Faktoren	4
■ die Steuerung eines Ökosystems durch abiotische Faktoren ...	4
■ die negativen Auswirkungen anaerober Verhältnisse am Gewässergrund/Phosphatfreisetzung aus dem Sediment	2
■ Bedeutung der Kläranlage	1
■ Einfluß der Temperatur auf die Ausprägung der sommerlichen Schichtung	1

Frage 6: Ich fand die Arbeit mit dem Computermodell:
(die Ziffern geben die Anzahl der Nennungen der jeweiligen Stufe an)

	++	+	0	+	++	
ermüdend	1	-	6	16	18	anregend
	++	+	0	+	++	
abwechslungsreich	14	20	6	2	-	monoton
	++	+	0	+	++	
anstrengend	3	7	22	4	4	einfach
	++	+	0	+	++	
interessant	25	14	2	-	-	langweilig
	++	+	0	+	++	
abstrakt	2	5	8	12	13	konkret
	++	+	0	+	++	
übersichtlich	23	13	3	3	-	unübersichtlich
	++	+	0	+	++	
anschaulich	28	3	1	-	-	unanschaulich
	++	+	0	+	++	
folgerichtig	22	8	2	-	-	zusammenhanglos

b) Bewertung:

Alle Schülerinnen und Schüler waren ausnahmslos der Meinung, daß ihr Vorwissen für die Arbeit mit dem Computermodell ausreichend war. Es ist daher anzunehmen, daß die im vorausgegangenen Unterricht lehrplangemäß vermittelten Kenntnisse und Einsichten die erforderlichen Grundlagen für ein Verständnis des Computermodells bereitstellen, also kein "Zusatzunterricht" notwendig wird. Auch der Schwierigkeitsgrad wird von fast allen Teilnehmern als "gerade richtig" eingeschätzt. Beides deutet auf eine gute Abstimmung der Anforderungsniveaus von Unterricht und Modell hin.

Die Antworten auf die Frage 3 des Schüler-Fragebogens zeigen eine überraschende Homogenität. Häufig werden Anschaulichkeit (20 Nennungen) und des öfteren auch Variabilität (9 Nennungen) als vorteilhafte Modelleigenschaften genannt. Viele Schülerinnen und Schüler schätzen das Computermodell als geeignet ein, Zusammenhänge aufzuzeigen (16 Nennungen). Diese Ergebnisse legen die Hypothese nahe, daß die für die Evaluation gestellten Fragen 1-3 (vgl. S. 112) positiv beantwortet werden können.

Einige Antworten beschäftigen sich mit dem didaktischen Ort für den Einsatz des Computermodells. Dieser Aspekt wurde auch in der Diskussion angesprochen. Die Schülerinnen und Schüler sind der Meinung, daß das Modell sehr geeignet für die Wiederholung, Vertiefung oder Zusammenfassung der Ergebnisse, aber weniger geeignet für eine reine Stoff- (Informations-) vermittlung ist.

Die Antworten auf die Frage 4 sind kontrovers. Teils verweisen die Schülerinnen und Schüler auf die Frage 3 oder wiederholen die bereits dort genannten Stichworte, teils gehen sie konkret auf *inhaltliche* Aspekte ein. Die Schüler bestätigen damit die in Abschnitt 2.3 (S. 18) erwähnte Schwierigkeit, zwischen "neuen Inhalten" und *neuen Zugängen* zu bereits vorhandenen Inhalten zu unterscheiden. Interessant sind vor allem die Antworten, das Modell zeige "dynamische Zusammenhänge", wogegen der traditionelle Unterricht mehr "starre Ausschnitte" böte (9 Nennungen), und "ermöglichte das Verfolgen eines Prozesses über längere Zeit" (5 Nennungen). Die Aussagen legen die Vermutung nahe, daß sich die Computersimulation durch diese didaktischen Eigenschaften von den traditionellen Medien unterscheidet. Antworten im Sinne "Das Verhalten eines Systems kann unter verschiedenen Bedingungen getestet werden" be-

stätigen eine der zentralen Zielsetzungen, durch die Bereitstellung vielfältiger Bearbeitungsmöglichkeiten eine qualifizierte Auseinandersetzung mit dem modellierten Sachverhalt zu erreichen.

Auf die Frage nach konkreten Zusammenhängen (Frage 5) wird vor allem der Einfluß des anthropogen bedingten Phosphatimports auf den Stoffkreislauf im See genannt. Insgesamt zeigen die detaillierten Antworten, daß die ausgewählten Beispiele für das Zusammenwirken der verschiedenen Faktoren in einem Ökosystem klar erkannt und reproduziert werden können. Das Computermodell scheint also die beabsichtigte Funktion der Veranschaulichung komplexer Zusammenhänge gut zu erfüllen. Die Auswertung der Frage 6 zeigt, daß die Schülerinnen und Schüler das Computermodell in allen aufgeführten Punkten überwiegend positiv einschätzen. Bei der sich an die schriftliche Befragung anschließenden Aussprache ergaben sich keine weiteren Gesichtspunkte für eine positive oder negative Bewertung des Modells.

Befragung von Biologielehrern und Seminarleitern

Zur Evaluation des Computermodells wurden weiterhin die Biologielehrer der Leistungskurse, sowie in einer gesonderten Diskussionsrunde, eine Gruppe von fünf Seminarleitern und Referendaren hinzugezogen. Im Mittelpunkt der Diskussion standen wiederum die auf Seite 112 genannten sechs Fragenschwerpunkte. Die Befragten beantworteten die Punkte 1 - 4 uneingeschränkt positiv. Was die Verständlichkeit der grafischen Darstellung anbelangt, so wurde von den Seminarleitern und Referendaren folgende Verbesserung angeregt: Wenn die Simulation in einem weiteren Modelljahr fortgesetzt wird, sollte bei der Grafikausgabe nicht der Eindruck entstehen, daß sich das Seebecken des Modellsees neu auffüllen würde. Dies könnte sonst zum Mißverständnis führen, die Simulation starte erneut mit den Anfangswerten. Diese Anregung wurde aufgegriffen und der Grafikaufbau entsprechend korrigiert. Die Gruppe bezeichnete weiterhin Ausdrucke mit einem grafikfähigen Matrixdrucker als unentbehrlich für die unterrichtliche Arbeit mit dem Modell. Erst durch sie kann eine vergleichende Analyse mehrerer Modellläufe und eine Auswertung der zahlreichen Details während der Modellläufe erfolgen. Auch diesem Vorschlag konnte inzwischen durch eine entsprechende Ergänzung des Computerprogramms entsprochen werden. Ansonsten wurde das Computerprogramm wegen seiner un-

mißverständlichen Bedienerführung als äußerst bedienungsfreundlich bezeichnet. Kontrovers diskutiert wurde abschließend die Frage, ob das ComputermodeLL neben der Vertiefung und Wiederholung des Unterrichtsstoffes auch zur Erarbeitung eingesetzt werden könnte. Dieser Aspekt hängt anscheinend sehr von der jeweiligen Gestaltung des Unterrichts ab und läßt sich nicht so ohne weiteres verallgemeinern. Eine Klärung ist letztlich nur durch eine gezielte praktische Erprobung herbeiführen.

6. Diskussion

Die rasante Entwicklung der Computertechnik führt zunehmend zu Herausforderungen auch an die didaktische Theorie und Praxis. Aus der Vielzahl der damit verbundenen Probleme wurden eingangs die zur Zeit wesentlichsten Fragen zu Voraussetzungen und Möglichkeiten der Computersimulation im Biologieunterricht aufgegriffen. Als eine der wichtigsten Antworten läßt die unterrichtliche Erprobung nunmehr die deutliche Tendenz erkennen, daß das Hauptziel bei der Entwicklung des Computermodells erreicht werden kann, nämlich das Zusammenwirken ausgewählter biotischer und abiotischer Faktoren im Ökosystem "See" zu veranschaulichen und dem Anwender damit eine Vorstellung von den vernetzten Zusammenhängen in einem Ökosystem zu vermitteln. Nach den bisherigen Unterrichtsanalysen ist es gelungen, das Ökosystem "See" im Modell einerseits soweit didaktisch zu vereinfachen, daß es in wesentlichen Verknüpfungen durchschaubar geworden ist, andererseits aber den Eindruck der Komplexität durch die zahlreichen Abhängigkeiten der verschiedenen Systemvariablen zu erreichen. Was die Modellgüte anbetrifft, so können die Reaktionen des Modells auf eine gezielte Änderung der zugänglichen Parameter als in hohem Maß plausibel bezeichnet werden.

Damit geben die Ergebnisse der unterrichtlichen Erprobung zu vorsichtigem Optimismus Anlaß, vorsichtig vor allem deswegen, weil Art und Umfang der Untersuchung, wie bereits erwähnt, noch nicht zu einer Generalisierung berechtigen. Auch kann der Einsatz des Computermodells keinesfalls losgelöst von der Gestaltung des Unterrichts gesehen werden. Aussagen über die didaktische Effizienz eines Mediums beziehen sich zwangsläufig auf den Kontext, in dem es eingesetzt wird und können damit nur mit Vorbehalt auf andersartige Unterrichtssituationen übertragen werden.

Die Ergebnisse der Evaluation zeigen weiterhin, daß das Computermodell *LAGO* in der getesteten Situation die vorgesehenen didaktischen Funktionen erfüllt: Es ermöglicht eine eigenständige Arbeit mit Modellen und zwingt durch die interaktive Arbeitsweise zu einer intensiven Auseinandersetzung mit dem Sachverhalt. Es genügt allen in Kapitel 3 genannten Anforderungen an Simulationsprogramme für den Biologieunterricht; insbesondere ist die Nutzung der spezifischen Fähigkeiten der verwendeten Rechner hervorzuheben, vor allem was die Rechenleistung und die grafischen Möglichkeiten angeht. Das Compu-

termodell kann daher als das eingangs geforderte, modellhafte Beispiel für ein Simulationsprogramm gelten, das die spezifischen Vorteile, die man sich vom Medium Computersimulation verspricht, zum Tragen bringt.

Ob nun die Computersimulation tatsächlich zusätzliche, lerneffektive Möglichkeiten für den Biologieunterricht eröffnet, kann letzten Endes erst durch breit angelegte empirische Untersuchungen erschöpfend geklärt werden. Mit dem Computermodell *LAGO* ist hierfür eines der notwendigen Instrumente geschaffen worden. Im Anschluß an diese Arbeit möchte ich daher eine solche Untersuchung, die einen Vergleich der unterrichtlichen Effizienz von Computersimulation und traditionellen Medien zum Thema hat, durchführen. Die im Rahmen dieser Arbeit beschriebene unterrichtliche Erprobung kann hierfür als "Erkundungsphase" (vgl. BORTZ 1985, S.4) gelten und damit eine Grundlage bei der Entwicklung weiterer Hypothesen bieten.

Für die Arbeit mit einem Computermodell kommen verschiedene Sozialformen des Unterrichts in Frage. Das Modell kann, wie bei der Erprobung geschehen, als "Demonstrationssimulation" mit der ganzen Klasse eingesetzt werden. Die Schüler können aber auch einzeln oder in kleinen Gruppen an verschiedenen Rechnern arbeiten, sofern die entsprechende technische Ausstattung vorhanden ist. Bei der letztgenannten Einsatzform ermöglicht der für die Arbeit mit dem Computer typische interaktive Dialog zwischen Programm und Anwender einen höheren Grad an Individualisierung und Differenzierung, vorausgesetzt das Computermodell bietet die hierfür erforderlichen vielfältigen und qualifizierten Arbeitsmöglichkeiten.

Das Computermodell *LAGO* hat rein deterministischen Charakter. Eine Schwäche solcher Modelle besteht darin, daß der Verlauf der Simulation durch die Wahl der Startbedingungen vorherbestimmt ist. Dieser Nachteil läßt sich ausgleichen, indem zusätzlich stochastische Einflüsse berücksichtigt werden. Ein deterministisches Modell kann aber aus didaktischer Sicht durchaus von Vorteil sein. Für den Schüler sind Vernetzungen biotischer und abiotischer Einflüsse leichter durchschaubar, wenn sich einzelne Faktoren eindeutig festlegen lassen. Bei Modellen mit stochastischem Charakter ist die Unterscheidung, welche Wirkungen aufgrund gezielt vorgenommener Parametereinstellungen zustandegekommen sind und welche zufallsbedingt waren, schwierig. Wenn dem Schüler allerdings die grundsätzlichen Zusammenhänge klar sind, kann die Berücksichtigung sto-

chastischer Elemente weiterführende Einsichten erbringen, da eine solche Erweiterung die Realitätsnähe eines Modells deutlich verbessert. Für das Modell *LAGO* ist daher folgende Weiterentwicklung vorgesehen: Da mit dem Rechner die Simulation stochastisch bedingten Verhaltens durch die zur Verfügung stehenden Zufallsgeneratoren leicht möglich ist, sollen einige Parameter durch Zufallsfunktionen mit wählbarer Verteilungsform ersetzt werden. Beispielsweise lassen sich für den Verlauf von Strahlung und Temperatur gestörte Sinuskurven verwenden. Um die genannte Schwierigkeit bei einem Modell, das stochastische Einflüsse berücksichtigt, auszuschalten, soll die Modellerweiterung so konzipiert werden, daß diese Einflüsse, vom Benutzer wählbar, an- und abgeschaltet werden können.

Das vorgestellte Verfahren zur Entwicklung von Computermodellen hat sich als sinnvoll und effizient erwiesen. Wie das Beispiel *LAGO* zeigt, können die Arbeiten in den Bereichen Didaktik, Limnologie, Mathematik und Programmieren so koordiniert werden, daß sich die im Mittelpunkt stehenden didaktischen Zielsetzungen konsequent realisieren lassen. Auf ein Problem möchte ich allerdings noch hinweisen: Der Gesamtaufwand der Entwicklungsarbeit übersteigt bei einem umfangreicheren Computermodell den für eine Einzelperson leistbaren Rahmen erheblich. Die vorliegende Dissertation stellt auch insofern eine Ausnahme dar, als Grundkenntnisse in den genannten Arbeitsbereichen in "Personalunion" teils vorhanden waren, teils erarbeitet werden konnten. Im Normalfall ist die Entwicklung qualifizierter Simulationsprogramme für den Unterricht aber nur im Team, bei arbeitsteiligem Verfahren, möglich.

Ein weiteres Problem, das die Softwareentwicklung in allen Anwendungsbereichen in Atem hält, sind die rasant fortschreitenden Leistungsverbesserungen auf dem Hardwaresektor. Für die Entwicklung von Simulationsprogrammen bringt dieser Prozeß teils Vorteile, teils zusätzliche Schwierigkeiten. Da die Leistungsfähigkeit solcher Programme in hohem Maß von der Leistungsfähigkeit der verwendeten Rechner abhängt, sind die Verbesserungen nur willkommen. Beispielsweise könnten dann beim Computermodell *LAGO* Kompartimente, auf die wegen der kritischen Ausführungsgeschwindigkeit verzichtet werden mußte, ergänzt werden, wie z.B. eine weitere Phytoplankton- oder Zooplanktongruppe, mit der sich die durch Konkurrenz und Selektion bedingte Sukzession in einem Ökosystem zeigen ließe. Zusätzliche Schwierigkeiten ergeben sich aber insofern, als die Entwicklung eines Simulationsprogramms nicht als abgeschlossen be-

trachtet werden kann. Um die möglicherweise in Frage kommenden Erweiterungen und Verbesserungen nicht unnötig zu erschweren, sollten sie bereits bei der Grundkonzeption mitbedacht werden. Ein Computerprogramm bedarf also der ständigen Programmpflege, zu der auch Anpassungen an eine veränderte Peripherie, wie z.B. Bildschirme mit verbesserter Auflösung oder neuentwickelte Drucker gehören.

Mit der Entwicklung des Computermodells und dem Optimismus, daß seine Verwendung im Biologieunterricht Vorteile bringen könnte, ist aber keinesfalls beabsichtigt, nun verstärkt den Einsatz von Simulationsmodellen zu fordern. Der Biologieunterricht zeigt leider ständig die bedenkliche Tendenz, in der Schulstube zu verharren und die Arbeit mit Medien in den Mittelpunkt zu stellen. Daß dies nicht erst seit heute ein Problem ist, zeigt der einprägsame Appell von Cornel Schmitt "Heraus aus der Schulstube !". Hierzu ließen sich noch viele weitere Beispiele aus der Geschichte des Biologieunterrichts nennen. Wenn nun die Computersimulation als weiteres, aufgrund seiner Aktualität besonderes Interesse weckendes Medium in den Unterricht drängt, wird diese Tendenz erneut verstärkt. "Nichts wäre verhängnisvoller, als wenn irreversibler und anerkannter technischer Fortschritt die nur zu Recht gezeißelte Zunft der Tafel- und Kreide-Biologen um scheinprogressive Bildschirm-Interpreten in der Gilde der Biologielehrer erweitern würde..." (LEHMANN/TISCHLER, 1987, S. 41). Im Mittelpunkt des Unterrichts steht die unmittelbare Naturbegegnung. Die Medien haben einerseits Ersatzfunktion, andererseits gibt es Fragestellungen, bei denen allein die Medien bestimmte Zugänge und Erkenntnismöglichkeiten erschließen.

Einer der Grundsätze für die Verwendung von Modellen im Unterricht lautet: Der Schüler darf das Modell nie für die Wirklichkeit halten. Wenn mit Computermodellen gearbeitet wird, muß umso mehr auf diesen Grundsatz geachtet werden, da der Umgang mit dem Computer Tendenzen zur Mystifizierung zeigt. Ergebnissen, die der Computer ausgibt, wird oft eine hohe Glaubwürdigkeit entgegengebracht. Der Biologieunterricht hat, wenn er ein komplexes Medium wie die Computersimulation einsetzt, auch eine erzieherische Aufgabe: Die Wirklichkeit, die durch den Computer hergestellt wird, muß in ihrer Reduktion und Entfremdung bewußtgemacht werden. Durch den Computer wird Wirklichkeit auf das reduziert, was sich mit dem Prinzip der binären Logik darstellen läßt. "Es ist eine eindeutige, datierbare, symbolische und simulierte Wirklichkeit,

die Persönlichkeit, menschliche Beziehungen, Sinnlichkeit und Leiblichkeit, Sinnhaftes und Qualität ausspart" (DANNER 1985). Die Computersimulation darf niemals zu einem beherrschenden Arbeitsmittel im Biologieunterricht werden, kann aber eine wertvolle Ergänzung und Erweiterung darstellen.

7. Zusammenfassung

Mit dieser Arbeit wurde ein Computermodell entwickelt, das beispielhaft einige der spezifischen Möglichkeiten der Computersimulation für den Biologieunterricht zum Tragen bringen soll. Die wesentlichen didaktischen Zielsetzungen, die der Modellentwicklung zugrunde lagen, sind:

- Das Modell soll ausgewählte Zusammenhänge im Ökosystem "See" veranschaulichen, wobei versucht wird, einerseits das reale System durch didaktische Vereinfachung durchschaubar zu machen, andererseits den Eindruck der Komplexität durch eine Vielzahl von Abhängigkeiten unter den verschiedenen Systemvariablen zu erhalten.
- Das Computermodell soll die spezifischen Möglichkeiten der Veranschaulichung von abstrakten Sachverhalten nutzen. Dazu gehört einerseits die Visualisierung durch eine übersichtliche grafische Darstellung der Struktur und des Verhaltens des Modell-Ökosystems. Veranschaulichung kann aber andererseits auch durch die aktive Auseinandersetzung mit dem Modell erreicht werden. Bei der Modellentwicklung müssen die hierfür erforderlichen vielfältigen Bearbeitungsmöglichkeiten geschaffen werden.
- Das Computermodell soll neben den für Medien allgemein geltenden Kriterien auch den spezifischen Anforderungen an Simulationsprogramme für den Biologieunterricht genügen. Besonderer Wert wurde bei der Entwicklung auf die Nutzung der spezifischen Fähigkeiten der verwendeten Rechner, vor allem was die Rechenleistung und die grafischen Möglichkeiten angeht sowie auf eine hohe Benutzerfreundlichkeit, gelegt.

Die Entwicklung eines Computermodells mit den genannten Zielsetzungen erweist sich als ein zeltaufwendiges umfangreiches Projekt, bei dem Arbeiten in mehreren, teilweise sehr unterschiedlichen Bereichen notwendig sind. Damit dieser Prozeß zielgerichtet und effizient erfolgen kann, wurde in der Arbeit ein für didaktische Zwecke geeignetes Verfahren aufgezeigt. Ausgehend von

einer Sachanalyse und einer didaktischen Analyse wird zunächst ein didaktisches Wortmodell erstellt, das eine genaue Beschreibung der erwünschten Struktur und des Modellverhaltens enthält. Aus dem Wortmodell wird ein formales Modell und schließlich ein Computerprogramm entwickelt. Um den Zeitaufwand bei der Programmierung zu reduzieren, wurden in der Arbeit einige wichtige Programmierhilfen genannt. Als letzter Schritt der Modellierung ist eine Evaluation des Computermodells erforderlich. Sie umfaßt die Überprüfung der Modellgüte und die Erprobung des Modells im Unterricht.

Das vorgestellte Verfahren wurde dann bei der Entwicklung des oben genannten Computermodells konkret angewendet. Das Modellökosystem setzt sich aus den Systemkomponenten Licht, Wassertemperatur, Orthophosphat, Phytoplankton, Zooplankton und Sauerstoff zusammen. Mit diesen Komponenten soll ein Modellverhalten erreicht werden, das Allgemeingültigkeitsansprüchen genügen kann. Struktur und Verhalten wurden im Wortmodell ausführlich beschrieben. Für die Erstellung des formalen Modells fiel die Entscheidung zugunsten der analytischen Modellierung. Kern des mathematischen Modells bildet ein Differentialgleichungssystem mit einer hohen Zahl von Rückkopplungen, das um trigonometrische Funktionen, Boolesche Funktionen und Tabellenfunktionen ergänzt wird. Als besondere, die Entwicklung von Simulationsmodellen für didaktische Zwecke kennzeichnende Schwierigkeit erwies sich die Notwendigkeit einer vergleichsweise extrem kurzen Ausführungszeit des Programms. Eine Lösung für dieses Problem wurde durch die Beschränkung des Modells auf wenige Zustandsgrößen und durch die Wahl einer dynamischen Schrittweitensteuerung bei der numerischen Approximation des Differentialgleichungssystems gefunden.

Bei der Überprüfung der Modellgüte konnte festgestellt werden, daß das Modellverhalten der im didaktischen Wortmodell festgelegten Beschreibung voll entspricht. Insgesamt kann man dem Modell eine hohe Plausibilität bescheinigen. Auch die unterrichtliche Erprobung ergab, daß das Modell die beabsichtigten didaktischen Zielsetzungen erfüllt.

LITERATUR

- ANDERSON, D.V.: A note on the morphology of the basins of the Great Lakes. J.Fish. Res. Bd. Canada, 18, 1961, S. 273-277
- BANNISTER, T.T.: Quantitative description of steady state, nutrient saturated algal growth, including adaptation. Limnol. Oceanogr. 24, 1979, S. 76-96
- BANOUB, G.: 19. Congress Inter. Verein. Limnolog., Winnipeg Can., 22. - 29. August in: KOZERSKI, H.-P.: Ein einfaches mathematisches Modell für den Phosphoraustausch zwischen Sediment und freiwasser. Acta hydrochim. hydrobiol. 5, 1977, 1, S. 53-65
- BARTH, G.: Computereinsatz in der Biologie - Humanbiologische Experimente. Bericht über die 8. Tagung der Fachleiter für Biologie, Kassel, 1985, gedrucktes Manuskript, Schriften des Verbandes zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts (MNU), Heft 34, 1985, S. 57-61
- BAYERISCHES LANDESAMT FÜR WASSERWIRTSCHAFT (Hrsg.): Zur Limnologie des Starnberger Sees. 3/82, München, 1982
- Beiträge zur Limnologie bayerischer Seen. 1/82, München, 1982
- BENNDORF, J.: Kausalanalyse, theoretische Synthese und Simulation des Eutrophierungsprozesses in stehenden und gestauten Gewässern. - Dissertation B, Technische Universität Dresden, 1979, S. 1-165
- The applicability of the Monod-model to the calculation of the phytoplankton growth rate in aquatic ecosystems. Proc. 7th Symp. Contin. Cultiv. Microorganisms, Academia, Prague 1980, pp. 305-312
- UHLMANN, D. and RECKNAGEL, F.: New aspects of the consideration of feedback mechanisms in an ecological model of the pelagic zone. Proc. Baden Symp., IAHS Publ., No. 125, 1978, S. 170-184
- BENNDORF, J. und RECKNAGEL, F.: Experimente mit einem dynamischen ökologischen Modell der Freiwasserregion von Talsperren und Seen. Acta hydrochim. hydrobiol., 7, 1979, 5, S. 473-490
- BERNHARDT, H.: Diskussionsbeitrag. Symposium "Eutrophierung und Gewässerschutz", Reinhardtbrunn, 16.-20. Oktober 1973
- BERTALANFFY, L. von /BEIER, W./LAUE R.: Biophysik des Fließgleichgewichts. Braunschweig, 1977
- BÖHRET, C./WORDELMANN, P.: Das Planspiel als Methode der Fortbildung. Köln, 1975
- BORTZ, J.: Lehrbuch der Statistik. 2. Aufl., Berlin/Heidelberg/New York/Tokyo, 1985
- BOSSEL, H.: Umweltdynamik. 30 Programme für kybernetische Umwelterfahrungen auf jedem BASIC-Rechner, München, 1985
- BRONSTEIN, I.N. and SEMENDJAJEW, K.A.: Taschenbuch der Mathematik. 19. Aufl. Thun und Frankfurt/M, 1980

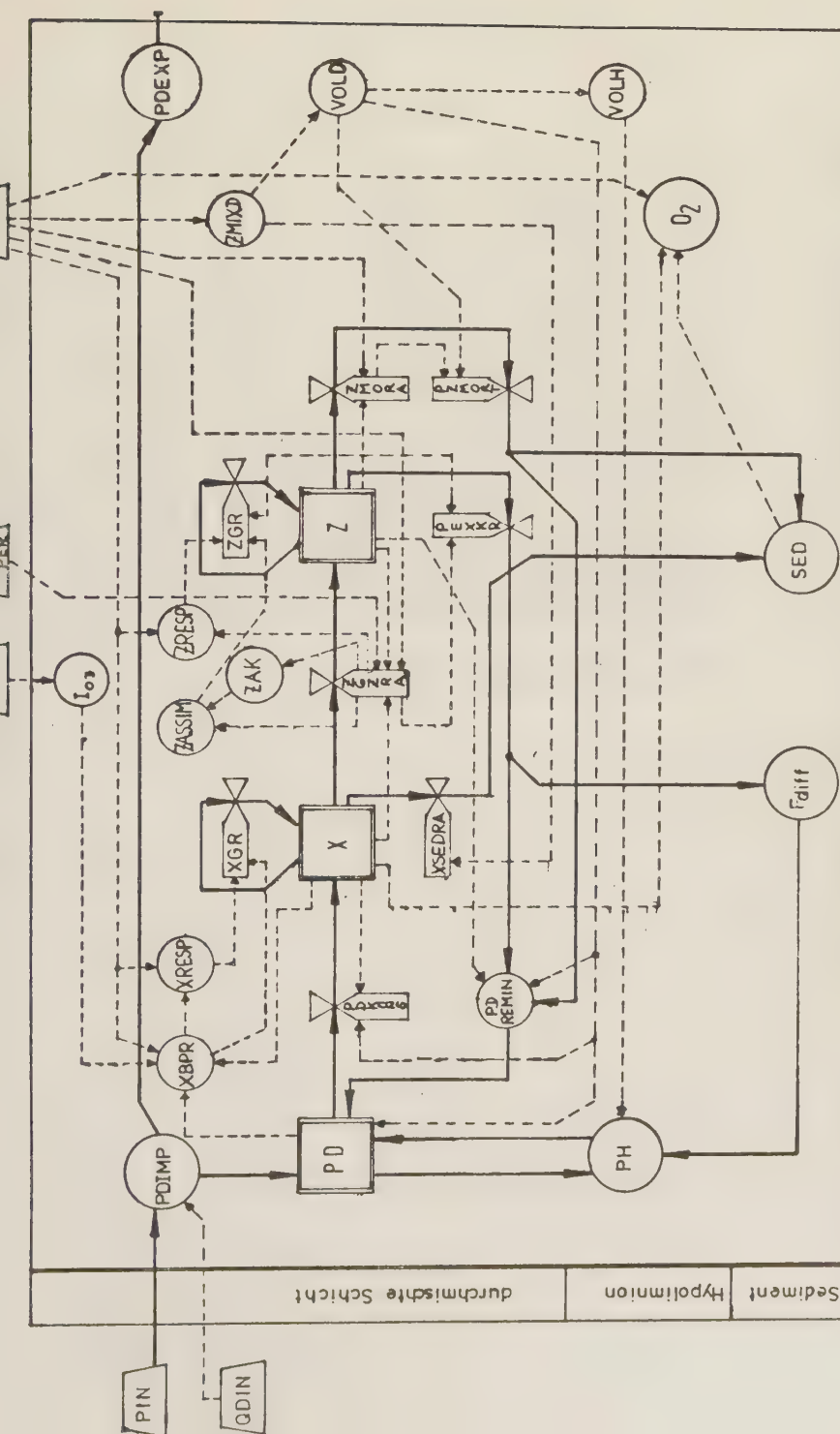
- BUND-LÄNDER-KOMMISSION FÜR BILDUNGSPLANUNG UND FORSCHUNGSFÖRDERUNG (BLK): Rahmenkonzept für die informationstechnische Bildung in Schule und Ausbildung. Drucksache K 43/84 vom 11.12.1984, Bonn, 1984
- BURNS, N.M. and ROSS C.: "Project Hypo". Discussion of Findings. Proc. 14th Conf. Great Lakes Res. 1971, S. 761-767
- CORNELSEN-VELHAGEN & KLASING (CVK): Das Programm für 1987; Realschulen, Gymnasien, Gesamtschulen. CVK-Computerprogramme, Lernsoftware Biologie, Berlin 1987, S. 17
- DANNER, H.: Kind, Computer und Wirklichkeit. In: Jahresheft Bildschirm, Seelze, 1985
- DI TORO, D.M., O'CONNOR, D.J. und THOMANN, R.V.: A dynamic model of phytoplankton populations in natural waters. Adv. Chem. Ser. 106, 1971, S. 131-180
- DEUTSCHER VEREIN ZUR FÖRDERUNG DES MATHEMATISCHEN UND NATURWISSENSCHAFTLICHEN UNTERRICHTS E.V.: Empfehlungen und Überlegungen zur Gestaltung von Lehrplänen für den Computer-Einsatz im Unterricht der allgemeinbildenden Schulen. Der mathematische und naturwissenschaftliche Unterricht (MNU), 38, Heft 4, 1985, S. 229-236
- DIEPOLD, P.: Kriterien für Unterrichts-Software. In: ARLT, W./ HAEFNER, K. (Hrsg.): Informatik als Herausforderung an Schule und Ausbildung, Berlin/Heidelberg/New York/Tokio, 1984, S. 169-174
- DVORAKOVA, M. und KOZERSKI, H.-P.: Three layer model of an aquatic ecosystem. ISEM Journal, 2, 1980, 1-2, S. 63-70
- EMSHOFF, J.R./SISSON, R.L.: Simulation mit dem Computer. München, 1972
- ESCHENHAGEN, D./KATTMANN, U./RODI, D.: Fachdidaktik Biologie, Köln, 1985
- EUROPÄISCHE GEMEINSCHAFTEN: Entschließung des Rates und der im Rat vereinigten Minister für Bildungswesen. Amtsblatt der Europäischen Gemeinschaften Nr. C 256/1 vom 24.9.1983, Brüssel, 1983
- FELS, G.: Zur Situation des Ökologieunterrichtes. Der Biologieunterricht, 17, 1981, 4, S. 5-26
- GARCON, V.: Modelling of Grangent Reservoir Ecosystem. In: Mitsch, W.J., Bossermann, R.W. and Klopatek, J.M. (Eds.): Energy and Ecological Modelling. Amsterdam 1981, pp. 293-304
- GENTIL, S.: A discrete model for the study of a lake. Appl. Math. Modelling, 3, 1979, June, S. 193-198
- GNAUCK, A.: Zur mathematischen Modellierung eines limnischen Ökosystems. Acta hydrochim. hydrobiol. 2, 1974, 6, S. 499-504
- GÖBEL, J., KOBLISCHKE, A.: Der Computer im Biologieunterricht. Unterricht Biologie, 5, 1981, Heft 60/61, 8/9, S. 51-57
- HARBORDT, S.: Computersimulation in den Sozialwissenschaften. Reinbek b. Hamburg, 1974
- HARTMANN, L. und BOES, M.: Mathematische Formulierung und Behandlung von Lebensgemeinschaften. Wasser- und Abwasserforschung, 5, 1973, S. 140-150
- HEIMSOETH SOFTWARE (Hrsg.): Turbo-Pascal 3.0. Handbuch. 2. Aufl., München, 1985

- HILLE, R. u.a.: Computer-Simulation und Modellbildung im Unterricht. Hochschuldidaktische Konzepte und Einsatzerfahrungen in den naturwissenschaftlichen Fächern. In: Simon, H., (Hrsg.): Computersimulation und Modellbildung im Unterricht. München/Wien, 1980
- HUTCHINSON, G.E.: Limnological studies in connecticut. IV. The mechanisms of intermediary metabolism in stratified lakes. Ecol. Monogr. 11, 1941, S.21-60
- A treatise on Limnology. Vol. 1-3, New York, 1957
- Eutrophication. The scientific background of a contemporary practical problem; American Scientist, 61, 1973, 5/6, S. 269-279
- IMBODEN, D.M.: Limnologische Transport- und Nährstoffmodelle. Schweiz. Z. f. Hydrol., 35, 1973, S. 29-68
- and GÄCHTER, R.: A dynamik lake model for trophic state prediction. Ecol. Modelling, 4, 1978, S.77-98
- JENSEN, K. and WIRTH, N.: Pascal User Manual and Report. Berlin, 1985
- JØRGENSEN, S.E.: A eutrophication model for a lake. Ecol. Modelling 2, 1976, S. 147-165
- Lake management. Pergamon Press, Oxford, 1979
- KAMP-NIELSEN, L. and JACOBSEN, O.S.: A submodel for anaerobic mud-water exchange of phosphate. Ecol. Modelling, 1, 1975, S.133-146
- MEYER, H. and FRIIS, M.: Examination of a lake model. Ecol. Modelling 4, 1978, S. 253-278
- - - JØRGENSEN, L. A. and HENDRIKSEN, J. (Eds.): Handbook of environmental data and ecological parameters. ISEM, Copenhagen, 1979
- KAISER, H.: Simulation in der Ökologie. Verhandlungen der Gesellschaft für Ökologie, Bremen, Band XIII, 1985, S.387-401
- KAMP-NIELSEN, L.: A kinetic approach to the aerobic sediment-water exchange of phosphorus in Lake Esrom. Ecol. Modelling, 1, S. 153-160
- KEIL, K.-A.: Programme für den Unterricht. Schulreport, Bayerisches Staatsministerium für Unterricht und Kultus (Hrsg.), 1985, Heft 1, S. 14-15
- KILLERMANN, W.: Biologieunterricht heute. 7. Aufl., Donauwörth, 1986
- KIRK, J.T.O.: A theoretical analysis of the contribution of algal cells to the attenuation of light within natural waters. II. Spherical cells. New Phytol. 75 (1975), S. 21-36
- KLAFFKI, W.: Didaktische Analyse als Kern der Unterrichtsvorbereitung. In: Auswahl 1, Hannover 1964, 4. Aufl., S. 5-34
- KLAPPER, H., ARNOLD, K.-H., BEUSCHOLDT, E., KOSCHEL, R., MOTHES, G.: Maßnahmen zum Schutz der Oberflächengewässer vor schädlichen Wirkungen aus der Mineraldüngung. Acta hydrochim. hydrobiol., 2, 1974, 3, S. 203-210
- KLAUS, G.: Wörterbuch der Kybernetik. Frankfurt, 1969

- KOLLER, H.: Simulation und Planspieltechnik. Wiesbaden, 1969
- KOSCHWITZ, H.: Der Mikrocomputer im Biologieunterricht. Der mathematisch-naturwissenschaftliche Unterricht, **38**, 1985, 8, S. 487-502
- KOZERSKI, H.-P.: Ein einfaches mathematisches Modell für den Phosphoraustausch zwischen Sediment und Freiwasser. Acta hydrochim. hydrobiol., **5**, 1977, 1, S. 53-65
- LAUDENBACH, B.: Der Computer als Medium im Biologieunterricht. Arbeitsberichte Folge 45' 86, Freie Hansestadt Bremen, Wissenschaftliches Institut für die Schulpraxis, Bremen, 1986
- LE CREN E. D. and LOWE-McCONNELL R.H. (Ed.): The functioning of freshwater ecosystems. Cambridge, New York, Melbourne, 1980
- LEHMAN, R. S.: Computer Simulation and Modeling, Hillsdale, 1977
- LEHMANN, D. und TISCHER, W.: Der Zauber Neuer Technologien im Biologieunterricht. Unterricht Biologie **122**, **11**, 1987, 2, S. 41-44
- LEVINS, R.: The strategy of model building in population biology. American Scientist, **54**, 1966, p. 421-431
- LIKENS, G.E. and BORMAN, F.M.: Linkages between terrestrial and aquatic ecosystems. Biosci. **24**, 1974, S. 447-456
- LÜKE, I./LÜKE, P.: Turbo-Pascal. Grundlagen und Anwendungen für alle Implementationen. München, 1985
- MANDL, C.E.: Simulationstechnik und Simulationsmodelle in den Sozial- und Wirtschaftswissenschaften. Berlin, 1977
- MCCREE, J. M.: An equation for the rate of respiration of white clover plants grown under controlled conditions. Proc. IBP/ DP Techn. Meet.: Prediction and measurement of photosynthetic productivity. CSAV, Trebon 1970, pp. 221-229
- MEADOWS, D. u.a.: Die Grenzen des Wachstums, Stuttgart, 1972
- MEYER, D./MEYER, G.: Das ökologische Problem der Räuber-Beute-Beziehung simuliert mit Hilfe eines Computerprogramms. Der Biologieunterricht, **11**, 1975, 2, S. 4-18
- MEYER, G.: Computersimulation von dynamischen Prozessen in Populationen. Unterricht Biologie, **8**, 1984, 8/9, S. 44-46
- Der Einsatz von Computern im Biologieunterricht. Kurzfassung eines einführenden Vortrags. Bericht über die 8. Tagung der Fachleiter für Biologie, Kassel 1985, gedrucktes Manuskript Schriften des Verbandes zur Förderung des mathematischen und naturwissenschaftlichen Unterrichts (MNU), Heft 34.
- MICHAEL, M.: Enhanced Graphics Adapter. Die Installation und Programmierung der IBM-HR- und kompatibler Grafikkarten, 2. Aufl., München, 1987
- MÖLLER, D.: Simulationen - Abbilder der Wirklichkeit. Computer Persönlich, Haar (Markt & Technik), 1986, Heft 4/86, S. 104 ff., Heft 5/86 S.98 ff., Heft 6/86 S.74 ff.

- MOSTLER, G./KRUMWIEDE, D./MEYER, G.: Methodik und Didaktik des Biologieunterrichts. 2. Aufl., Heidelberg, 1979
- NAUMANN, E.: Nägra synpunte angående planktons ökologi. Med. sarskild hänsyn Aill fytöplankton. Svensk. bot. Tidskr. 13, 1919, S. 129-158
- NEUMANN, I.: Maximum depth and average depth of lakes. J. Fish. Res. Bd. Canada, 16, 1959, S. 923-927
- ODUM, P.: Grundlagen der Ökologie, Bd.1: Grundlagen. Bd.2: Standorte und Anwendung. Stuttgart/New York, 2. Aufl. 1983
- OHLE, W: Gelöste organische Stoffe, Aufnahme und Abgabe durch Phytoplanktonorganismen im See. Gewässerschutz-Wasser-Abwasser, 8, 1972, S. 1-56
- PETERS, R.H. and RIGLER, F.H.: Phosphorus release by Daphnia. Limnol. Oceanogr. 18, 1973, S. 821-839
- REICHART, G.: Perlenspiel und Computer-Simulation zum Hardy-Weinberg-Gesetz. Praxis der Naturwissenschaften Biologie 33, 1984, H. 1, S. 24-26
- REICHOLF, J.: Der See als Ökosystem und die Rolle der Wasservögel. Tagungsbericht der Akademie für Naturschutz und Landschaftspflege (Hrsg.) Laufen/Salzach, Nr.3/79, S. 69-82
- REINISCH, K.: Kybernetische Grundlagen und Beschreibung kontinuierlicher Systeme. Berlin, 1974
- REMMERT, H.: Ökologie - ein Lehrbuch. Berlin/Heidelberg/New York/Tokio, 3. Aufl., 1984
- RICHTER, O.: Simulation des Verhaltens ökologischer Systeme. Mathematische Methoden und Modelle, Weinheim, 1985
- SCHAEFER, G.: Grundsätze zu einer Didaktik der Ökologie. In: RIEDEL, W. und TROMMER, G. (Hrsg.): Didaktik der Ökologie, Köln, 1981, S. 18-46
- SCHELLENBERGER, G./BEHRENDT, H./KOZERSKI, H.-P., MOHAUPT, V.: Ein mathematisches Ökosystemmodell für eutrophe Flachgewässer. Acta Hydrophysica, 28, 1/2, 1983, S. 109-172
- SCHMIDT, E.: Ökosystem See. Das Beziehungsgefüge der Lebensgemeinschaft im eutrophen See und die Gefährdung durch zivilisatorische Eingriffe. Biologische Arbeitsbücher, Band 12, Heidelberg, 1974
- SCHOOF, J.: Biologie auf dem Apple II. Braunschweig, 1986, S. 97-109
- SCHUBERT, R. (Hrsg.): Lehrbuch der Ökologie. Jena, 1984
- SCHWOERBEL J.: Einführung in die Limnologie. 5. Aufl., Stuttgart, 1974
- SEITZ, A.: Simulationsprogramme - Techniken und Anwendungen in der Populationsbiologie. Schriftliche Leistung zur Erlangung des akademischen Grades Dr. rer. nat. habil. an der Fakultät für Biologie, Chemie und Geowissenschaften der Universität Bayreuth, 1985
- SIEBECK, O.: Der Königsee. Eine limnologische Projektstudie. Arbeitsgruppe Limnologie des Zoologischen Instituts der Universität München, Leitung und Ausarbeitung: Prof. Dr. O. SIEBECK. Hrsg.: Nationalparkverwaltung Berchtesgaden, im Auftrag des Bayerischen Staatsministeriums für Landesentwicklung und Umweltfragen. Forschungsberichte 5/1982

- SIMON, H.: (Hrsg.): Computersimulation und Modellbildung im Unterricht, München/Wien, 1980
- STRASKRABA, M.: The effect of physical variables on freshwater production: Analyses based on models. In: Le CREN, E.D., and LOWE-McCONNELL, R.H. (Eds.): The functioning of freshwater ecosystems. Cambridge Univ. Press, Cambridge, 1980, pp. 13-84
- und DVORAKOVA, M.: Probleme der Simulation von Talsperrenökosystemen, Acta hydrochim. hydrobiol., 5, 1977, 3, S. 211-230
- und GNAUCK, A.: Freshwater Ecosystems. Modelling and Simulation. Developments in environmental modelling, 8, Jena, 1985
- THIENEMANN, A.: Untersuchungen über die Beziehungen zwischen dem Sauerstoffgehalt des Wassers und der Zusammensetzung der Fauna in norddeutschen Seen. Arch. Hydrobiol., 12, 1918, S. 1-65
- Der Bau des Seebeckens in seiner Bedeutung für den Ablauf des Lebens im See. Verh. Zool.-Bot. Ges. Wien, 77, 1927, S. 87-91
- TISCHLER, W.: Einführung in die Ökologie. Stuttgart/New York (Gustav Fischer Verlag), 3.Aufl., 1984
- TROMMER, G.: Ökologie. Köln, 1983
- VESTER, F.: Unsere Welt - ein vernetztes System. Eine internationale Wanderausstellung von Frederic Vester. Stuttgart, 1978
- VOLLENWEIDER, R.A.: Möglichkeiten und Grenzen elementarer Modelle der Stoffbilanz von Seen. Arch. Hydrobiol. 66, 1969, S. 1-36
- WEBER, C.A.: Aufbau und Vegetation der Moore Norddeutschlands. Beibl. Bot. Jahrb. 90, (Suppl. to Bot. Jahrb. 40), 1907, S. 19-34
- WEDEKIND, J.: Unterrichtsmedium Computersimulation. Eine mediendidaktische Standortbestimmung mit Unterrichtsbeispielen aus der Biologie. Neue Lehrverfahren, Bd.11, Prof. Dr. Walther Zifreund, (Hrsg.), Weil der Stadt/Württ., 1981
- WETZEL R. G.: Limnology. Philadelphia/Toronto, 1975
- WINDE, P.: Modelle im Ökologieunterricht. In: RIEDEL, W. und TROMMER, G. (Hrsg.): Didaktik der Ökologie, Köln, 1981, S. 277-298
- Zur Funktion der Computersimulation in der Umwelterziehung. Log In 3, 1983, H.1, S. 16-20
- Zur Erweiterung der Modellmethode durch die Computersimulation. Biologica didactica, 6, Heft 4, 1984, S. 27-35



Anlage 1: Darstellung der Struktur des Computermodells LAGO unter Berücksichtigung der wichtigsten Systemgrößen und ihrer Wechselwirkungen (Erläuterung der Symbole siehe nächste Seite)

Erläuterung der Symbole und Abkürzungen:



Zustandsgröße



Veränderungsrate



Input-Variable



Hilfsvariable



Masse- und Energiefluß

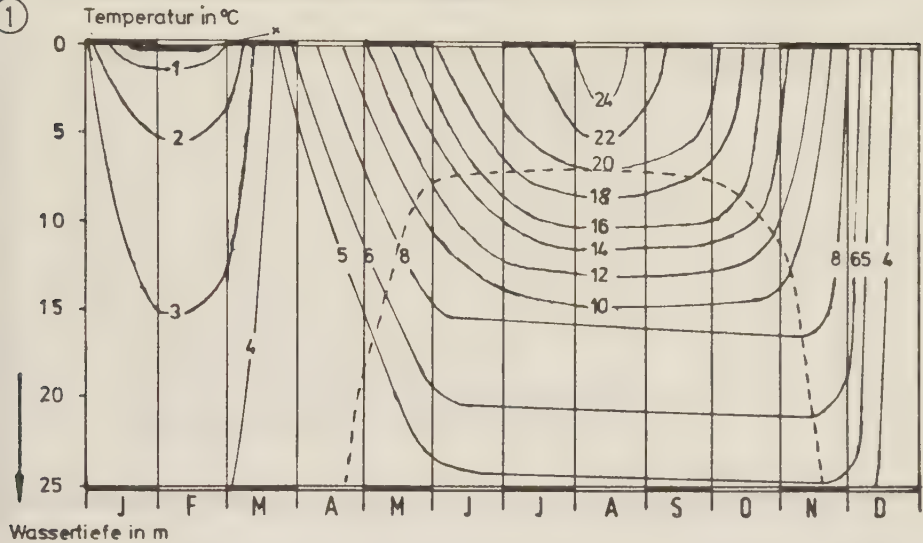


Informationsfluß

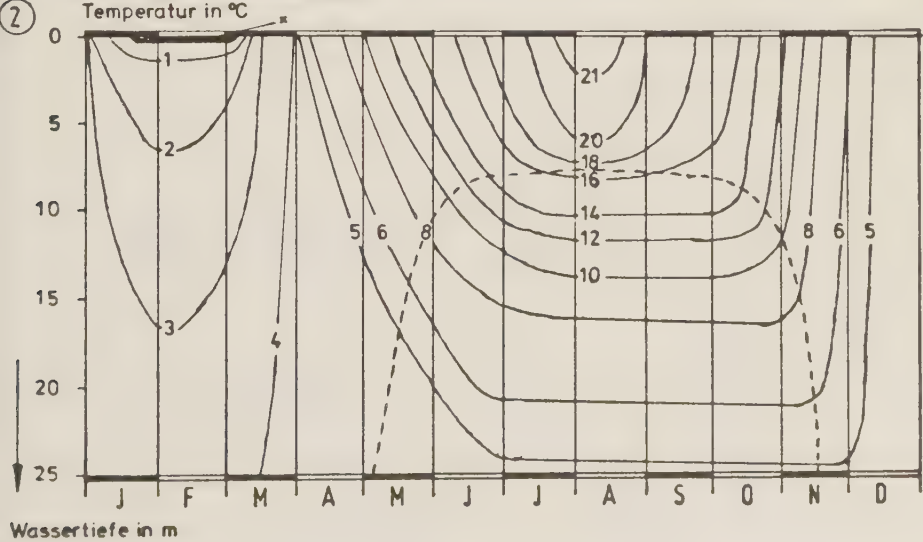
Fdiff	Phosphatfreisetzung aus dem Sediment [$\text{mg P} \cdot \text{m}^{-2} \cdot \text{d}^{-1}$]
I	Licht (400...700 nm) über der Gewässeroberfläche [$\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}$]
I_{0z}	Licht (400...700 nm) in der durchmischten Schicht [$\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}$]
O₂	Sauerstoffkonzentration [mg/l]
PD	Orthophosphatkonzentration in der durchmischten Schicht [mg P/m^3]
PDEXP	Phosphatexport für durchmischte Schicht [mg P/d]
PDIMP	Phosphatimport für durchmischte Schicht [mg P/d]
PDKONS	Phosphatverbrauch durch Phytoplanktonwachstum in der durchmischten Schicht [mg P/d]
PDREXTN	Phosphatremineralisation in der durchmischten Schicht [mg P/d]

PEXRR	Rate der Phosphatremineralisation durch Exkretion des Zooplanktons [$\text{mg P} \cdot \text{cm}^{-3} \cdot \text{d}^{-1}$]
PH	Orthophosphatkonzentration im Hypolimnion [mg P/m^3]
PHOTPER	Lichtstunden pro Tag [h]
PHN	Orthophosphatkonzentration im Zufluß [mg P/m^3]
PZWORT	Rate der Phosphatremineralisation aus abgestorbenen oder von Fischen gefressenen Zooplankton [$\text{mg P} \cdot \text{cm}^{-3} \cdot \text{d}^{-1}$]
QDIN	Wasserzufluß in die durchmischte Schicht [m^3/d]
SED	am Gewässergrund sedimentierende Biomassen [d^{-1}]
TEMP	mittlere Wassertemperatur in ca. 5 m Tiefe [$^{\circ}\text{C}$]
VOLH	Volumen des Hypolimnions [m^3]
VOLD	Volumen der durchmischten Schicht [m^3]
X	Phytoplanktonbiomasse (-dichte) [cm^3/m^3]
XBPR	Brutto-Photosyntheserate des Phytoplanktons in der durchmischten Schicht [d^{-1}]
XGR	Wachstumsrate in der durchmischten Schicht (Nettoproduktion) [d^{-1}]
XRESP	Respirationsrate des Phytoplanktons in der durchmischten Schicht [d^{-1}]
XSEDA	Sedimentationsrate des Phytoplanktons [d^{-1}]
Z	Zooplanktonbiomasse (-dichte) [cm^3/m^3]
ZAK	Verhältnis der assimilierten zur gefressenen Nahrungsmenge des Zooplanktons [cm^3/cm^3]
ZASSTM	Assimilationsrate des Zooplanktons [d^{-1}]
ZGR	Wachstumsrate des Phytoplanktons [d^{-1}]
ZGZRA	Grazing-Rate des Zooplanktons [d^{-1}]
ZMIXD	Tiefe der durchmischten Schicht [m]
ZMORA	Mortalitätsrate des Zooplanktons [d^{-1}]
ZRESP	Respirationsrate des Zooplanktons [d^{-1}]

warmer Sommer/warmer Winter/wenig Bewölkung

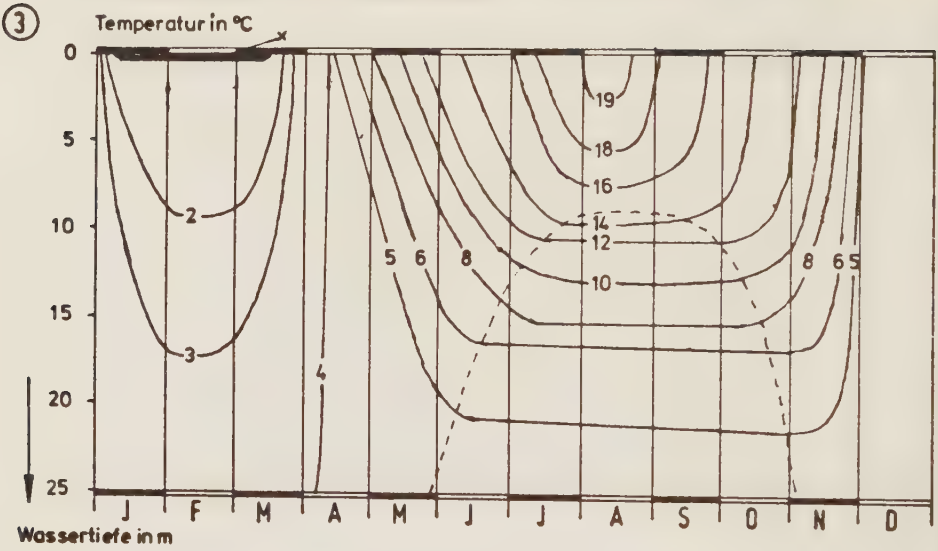


normaler Sommer/normaler Winter



x: Eis- und Schneedecke
gestrichelte Linie: Obergrenze Metalimnion

kühler Sommer/kalter Winter/häufige Bewölkung



Anlage 3

Mathematisches Modell

Zustandsvariablen

X	Phytoplanktonbiomasse (-dichte) [cm^3/m^3]
Z	Zooplanktonbiomasse (-dichte) [cm^3/m^3]
PD	Orthophosphatkonzentration in der durchmischten Schicht [$\text{mg P}/\text{m}^3$]
PH	Orthophosphatkonzentration im Hypolimnion [$\text{mg P}/\text{m}^3$]

Zustandsgrößen

$\frac{dX}{dt}$	=	$XG - X\text{SEDI} - ZGZ$ [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]
$\frac{dZ}{dt}$	=	$ZG - Z\text{MORT}$ [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]
$\frac{dPD}{dt}$	=	$(PD\text{IMP} - PD\text{EXP} - PDKONS + PD\text{REMIN})$ [$\text{mg P} \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]

Störgrößen

I Licht (400 ... 700 nm) über der Gewässeroberfläche [$\text{J} \cdot \text{cm}^{-2} \cdot \text{d}^{-1}$]

$$I = \{ k_I \cdot 8,0 \cdot [280 + 220 (\sin(t + 280))] / 24 \} \cdot \text{PHOTPER}$$

warmer Sommer/warmer Winter/wenig Bewölkung: $k_I = 1,3$

normaler Sommer/normaler Winter: $k_I = 1,0$

kühler Sommer/kalter Winter/häufige Bewölkung: $k_I = 0,7$

$$k_I = 29,0$$

PHOTPER Lichtstunden pro Tag [h]

$$\text{PHOTPER} = 12 - 4 \cdot \cos t$$

P'_i Fracht der i-ten phosphatbelasteten Quelle [g P/d]

$$P'_i = SBK_i \cdot E_i$$

$SBK_1 = 2,5$ [g P/d] bei 1000 mm Niederschlag/a (E_1)

$SBK_2 = 2,3$ [g P/d] pro Einwohner (E_2)

$SBK_3 = 1,7$ [g P/d] pro Einwohner (E_3)

$SBK_4 = 3,8$ [g P/d] pro ha landwirtsch. Fläche (E_4)

PIN Orthophosphatkonzentration im Zufluß [$\mu\text{g P/m}^3$]

$$PIN = \sum_{i=1}^i P'_i / QDIN$$

$QDIN$ Wasserzufluß in die durchmischte Schicht [m^3/d]

$$QDIN = QDOUT$$

$QDOUT$ Wasserabfluß aus der durchmischten Schicht [m^3/d]

$$QDOUT = T_w \cdot V$$

$$T_w = 720 \text{ [d]}$$

$TEMP$ mittlere Wassertemperatur in ca. 5 m Tiefe [$^{\circ}\text{C}$]

$$TEMP = (TK1 + TK2) + TK1 \cdot \sin(t + 225)$$

warmer Sommer/warmer Winter/wenig Bewölkung: $TK1 = 10,5$; $TK2 = 1,5$

normaler Sommer/normaler Winter: $TK1 = 9,0$; $TK2 = 1,5$

kühler Sommer/kalter Winter/häufige Bewölkung: $TK1 = 8,0$; $TK2 = 0,5$

Umgebungsgrößen

z_m maximale Tiefe [m]

$$z_m = 25 \text{ [m]}$$

z_r relative Tiefe [m]

$$z_r = \frac{50 \cdot z_m \cdot \sqrt{n}}{\sqrt{A}}$$

z' normierte Tiefe [dimensionslos]

$$z' = ZMIXD / z_m$$

$\bar{z}$ mittlere Tiefe [m]

$$\bar{z} = A / V$$

ZMIXD Tiefe der durchmischten Schicht [m]

warmer Sommer/warmer Winter/wenig Bewölkung:

Dekade 1..9 oder 34..36: ZMIXD = z_m

Dekade 10 oder 32..33: ZMIXD = 16 [m]

Dekade 11..12 oder 29..31: ZMIXD = 12 [m]

Dekade 13..28: ZMIXD = 5 [m]

normaler Sommer/normaler Winter:

Dekade 1..9 oder 34..36: ZMIXD = z_m

Dekade 10..11 oder 30..33: ZMIXD = 16 [m]

Dekade 12..13 oder 28..29: ZMIXD = 12 [m]

Dekade 14..27: ZMIXD = 7 [m]

kühler Sommer/kalter Winter/häufige Bewölkung:

Dekade 1..11 oder 34..36: ZMIXD = z_m

Dekade 12..13 oder 32..33: ZMIXD = 20 [m]

Dekade 14..16 oder 29..31: ZMIXD = 16 [m]

Dekade 17..28: ZMIXD = 12 [m]

p Transformationsparameter zur näherungsweisen Ermittlung des Volumens des Wasserkörpers (extremes elliptisches Sinusoid) bei jeder gewünschten Wassertiefe ZMIXD [dimensionslos]

$$p = 6 (\bar{z} / z_m) - 3$$

A Fläche des Sees (elliptische Oberfläche) [m²]

$$A = \pi \cdot a \cdot b$$

$a = 2000$ [m]

$b = 1333$ [m]

V Volumen des gesamten Wasserkörpers [m³]

$$V = 4 (1 - 2/\pi) \cdot a \cdot b \cdot z_m$$

V_z' Volumen des Wasserkörpers in der normierten Tiefe z' [dimensionslos]

$$V_z' = [6 z' - (1 - p) \cdot ZMIXD^2 - 2 p \cdot ZMIXD^3] / (3 + p)$$

VOLD Volumen der durchmischten Schicht [m^3]

$$VOLD = V_z' \cdot V$$

VOLH Volumen des Hypolimnions [m^3]

$$VOLH = V - VOLD$$

Veränderungsraten

PKONS Phosphatverbrauch durch Phytoplanktonwachstum in der durchmischten Schicht [$mg\ P/d$]

$$PKONS = XGR / KKNUP \cdot X \cdot VOLD$$

$KKNUP = 0.8$; (kann vom Benutzer verändert werden)

PDREMIN Phosphatremineralisation in der durchmischten Schicht [$mg\ P/d$]

$$PDREMIN = (PZMORT + PEXKR) \cdot Z \cdot VOLD$$

PEXKR Rate der Phosphatremineralisation durch Exkretion des Zooplanktons
[$mg\ P \cdot cm^{-3} \cdot d^{-1}$]

$$PEXKR = 0,5 \cdot \left[\frac{ZGZRA}{KKNUP} - \frac{ZGR}{KZNUF} \right]$$

$KKNUP = 0,8$; (kann vom Benutzer verändert werden)

$KZNUF = 0,8$; (kann vom Benutzer verändert werden)

PHVZ Phosphatkonzentration zu Beginn der Herbstvollzirkulation [$mg\ P/m^3$]

$$PHVT(t-\Delta t) = \frac{PD(t-\Delta t) \cdot VOLD(t-\Delta t) + PH(t-\Delta t) \cdot VOLH(t-\Delta t)}{VOLD(t-\Delta t) + VOLH(t-\Delta t)}$$

PZMORT Rate der Phosphatreminalisation aus abgestorbenen oder von Fischen gefressenem Zooplankton [$\mu\text{g P} \cdot \text{cm}^{-3} \cdot \text{d}^{-1}$]

$$\text{PZMORT} = 0,5 \cdot \text{ZMORA} / \text{KZNU}$$

KZNU = 0,8; (kann vom Benutzer verändert werden)

XBPR Brutto-Photosyntheserate des Phytoplanktons in der durchmischten Schicht [d^{-1}]

$$\text{XBPR} = \left[\text{XBPRMIN} + \left(\frac{\text{XBPRMAX} - \text{XBPRMIN}}{\text{XGTOPT}} \right) \cdot \text{TEMP} \right] \cdot \frac{\text{P/X}}{\text{KXP/KX} + \text{P/KX} + \text{KXP/X} + \text{P/X}} \cdot \text{I}_{02}$$

$$\text{KXP} = 1,7$$

$$\text{XBPRMAX} = 1,8$$

$$\text{XBPRMIN} = 0,2$$

XGTOPT = 20,0; (kann vom Benutzer verändert werden)

XG Phytoplanktonwachstum [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]

$$\text{XG} = \text{XGR} \cdot \text{X}$$

XGR Wachstumsrate in der durchmischten Schicht (Nettoproduktion) [d^{-1}]

$$\text{XGR} = \text{XBPR} - \text{XRESP}$$

XRESP Respirationsrate des Phytoplanktons in der durchmischten Schicht [d^{-1}]

$$\text{XRESP} = \text{XRPP} + 0,3 \cdot \text{XBPR}$$

XSEDI Phytoplanktonverluste durch Sedimentation [d^{-1}]

$$\text{XSEDI} = \text{XSEDRA} \cdot \text{X}$$

XSEDRA Sedimentationsrate des Phytoplanktons [d^{-1}]

$$\text{XSEDRA} = \text{XS} / \text{ZMIXD}$$

XS = 0,1; (kann vom Benutzer verändert werden)

ZAK Verhältnis der assimilierten zur gefressenen Nahrungsmenge des Zooplanktons [cm^3/cm^3]

$$ZAK = \left[ZAKMAX - \left(\frac{ZAKMAX - ZAKMIN}{ZGZMAX} \right) \cdot ZGZRA \right] \cdot UXZ$$

UXZ = 1,0; (kann vom Benutzer verändert werden)

ZAKMAX = 0,8

ZAKMIN = 0,4

ZGZMAX = 2,5

ZASSIM Assimilationsrate des Zooplanktons [d^{-1}]

$$ZASSIM = ZAK \cdot ZGZRA$$

ZG Zooplanktonwachstum [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]

$$ZG = ZGR \cdot Z$$

ZGR Wachstumsrate des Zooplankton [d^{-1}]

$$ZGR = ZASSIM - ZRESP$$

ZGZ Phytoplanktonverluste durch Freßtätigkeit (Grazing) des Zooplanktons [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]

$$ZGZ = ZGZRA \cdot Z$$

ZGZDU Grazing-Rate bei Dunkelheit [d^{-1}]

$$ZGZDU = \left\{ 2,25 \cdot \left[\exp \left(-2,0 \cdot \text{abs} \left(\ln \frac{\text{TEMP}}{ZGZTOPT} \right) \right) \right] + 0,25 \right\} \cdot \frac{X/Z}{KGZX/KZ + X/KZ + KGZX/Z + X/Z}$$

KGZX = 5,0

ZGZTOPT = 20,0; (kann vom Benutzer verändert werden)

ZGZLI Grazing-Rate bei Licht [d^{-1}]

$$\text{ZGZLI} = \text{ZGZDU} \cdot \left(\frac{24 - \text{PHOTPER}}{24} \right) + \text{ZGZI} \cdot \text{ZGZDU} \cdot \frac{\text{PHOTPER}}{24}$$

$$\text{ZGZI} = 0,33$$

ZGZRA Grazing-Rate des Zooplanktons [d^{-1}]

$$\text{ZGZRA} = \text{ZGZDU} \text{ bei } \text{HVZ} \leq t < \text{SST}$$

$$\text{ZGZRA} = \text{ZGZLI} \text{ bei } \text{SST} \leq t < \text{HVZ}$$

ZMORA Mortalitätsrate des Zooplanktons [d^{-1}]

$$\text{ZMORA} = (\text{ZMOMIN} + \text{ZMOT} \cdot \text{TEMP}) \cdot \frac{Z}{\text{KZMO} + Z}$$

$$\text{KZMO} = 0,35$$

$$\text{ZMOMIN} = 0,01$$

$$\text{ZMOT} = 0,004$$

ZMORT Zooplanktonmortalität [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]

$$\text{ZMORT} = \text{ZMORA} \cdot Z$$

ZRESP Respirationsrate des Zooplanktons [d^{-1}]

$$\text{ZRESP} = \left[\left(\frac{\text{ZRESPOPT} - \text{ZRESPMIN}}{\text{ZGZMAX}} \right) \cdot \text{ZGZRA} + \text{ZRESPMIN} \right] \cdot \frac{1}{\text{ZRESPOPT}}$$

$$\cdot \left[(\text{ZRESPOPT} - \text{ZRSPTMIN}) \cdot \left(\frac{\text{TEMP}}{\text{ZGZTOPT}} \right)^2 + \text{ZRSPTMIN} \right]$$

$$\text{ZGZMAX} = 2,5$$

$$\text{ZGZTOPT} = 20,0; \text{ (kann vom Benutzer verändert werden)}$$

$$\text{ZRESPMIN} = 0,1$$

$$\text{ZRESPOPT} = 0,05$$

$$\text{ZRSPTMIN} = 0,05$$

Zwischengrößen

CUMSED am Gewässergrund sedimentierende Biomassen [d^{-1}]

$$CUMSED(t) = CUMSED(t-t) + XSEDI/3 + ZMORT/3$$

$$CUMSED = 0,7 \cdot CUMSED \quad \text{für } t = \text{FVZ}$$

$$CUMSED = 8,85 \cdot CUMSED \quad \text{für } t = \text{HVZ}$$

ϵ Extinktionskoeffizient für Licht (400 ... 700 nm) [dimensionslos]

$$\epsilon = EPSAQUA - (LITE - XEXT \cdot X) / 2$$

$$EPSAQUA = 0,95$$

$$XEXT = 0,2$$

F_{diff} Phosphatfreisetzung aus dem Sediment [$mg \text{ P} \cdot m^{-2} \cdot d^{-1}$]

$$F_{diff} = 1.21 (P_i - P_H) - 1.70$$

FTE Faktor für Trübstoffeintrag [dimensionslos]

$$FTE = 5 \cdot [1,5 + 0,5 \cdot \cos(t - 30)]$$

FVZ Nummer der Startdekade für Frühjahrsvollzirkulation

$$\text{warmer Sommer/warmer Winter/wenig Bewölkung: } SST = 8$$

$$\text{normaler Sommer/normaler Winter: } SST = 10$$

$$\text{kühler Sommer/kalter Winter/häufige Bewölkung: } SST = 13$$

I_0 durch Reflexion reduziertes Licht (400 ... 700 nm) direkt unter der Wasseroberfläche [$J \cdot cm^{-2} \cdot d^{-1}$]

$$I_0 = RI \cdot I$$

$$RI = 0,2 \text{ wenn Eis- und Schneedecke vorhanden, sonst}$$

$$RI = 0,9$$

I_{0z} Licht (400 ... 700 nm) in der durchmischten Schicht [$J \cdot cm^{-2} \cdot d^{-1}$]

$$I_{0z} = \frac{\ln(KI + I_0) - \ln\{KI + [I_0 \cdot \exp(-\epsilon \cdot ZMIXD)]\}}{\epsilon \cdot ZMIXD}$$

KX Halbsättigungswert für die inverse Beziehung zwischen Photosyntheserate und Biomasse des Phytoplanktons [dimensionslos]

$$KX = KXP1 \cdot PDKXP2$$

$$KXP1 = 2,8$$

$$KXP2 = 0,4$$

KZ Halbsättigungswert für die inverse Beziehung zwischen Grazing-Rate und Biomasse des Zooplanktons [dimensionslos]

$$KZ = KZP1 \cdot XKZP2$$

$$KZP1 = 5,7$$

$$KZP2 = 0,4$$

LTE Logarithmus der prozentualen Lichttransmission bei 5 m Schichtdicke (Epilimnion) [dimensionslos]

$$LTE = 1,2$$

$$\text{falls } FTE \leq FTEMAX$$

$$LTE = 1,2 - [0,6 \cdot (FTE - FTEMAX)] \quad \text{falls } FTE > FTEMAX$$

$$FTEMAX = 4,3$$

PDEXP Phosphatexport für durchmischte Schicht [mg P/d]

$$PDEXP = PD \cdot QDOUT$$

PDIMP Phosphatimport für durchmischte Schicht [mg P/d]

$$PDIMP = PIN \cdot QDIN$$

Die Größe PIN wird vom Benutzer festgelegt.

SST Nummer der Startdekade für Sommerstagnation

$$\text{warmer Sommer/warmer Winter/wenig Bewölkung: } SST = 13$$

$$\text{normaler Sommer/normaler Winter: } SST = 14$$

$$\text{kühler Sommer/kalter Winter/häufige Bewölkung: } SST = 17$$

HVZ Nummer der Startdekade für Herbstvollzirkulation

$$HVZ = 33$$

XRPP Grundumsatz der Phytoplanktonrespiration [d^{-1}]

$$XRPP = 0,02 \cdot TEMP$$

Tabellen

Löslichkeit von Sauerstoff in reinem Wasser in Abhängigkeit von der Wassertemperatur (bei sauerstoffgesättigter Luft und einem Luftdruck von 760 Torr):

Temperatur	Sauerstoff- konzentration	Temperatur	Sauerstoff- konzentration
°C	mg/l	°C	mg/l
0	14,16	14	9,98
1	13,77	15	9,76
2	13,40	16	9,56
3	13,05	17	9,37
4	12,70	18	9,18
5	12,37	19	9,01
6	12,06	20	8,84
7	11,76	21	8,68
8	11,47	22	8,53
9	11,19	23	8,38
10	10,92	24	8,25
11	10,67	25	8,11
12	10,43	26	7,99
13	10,20	27	7,86

aus: WETZEL 1975, S. 124

Zusammenhang zwischen sedimentierender Biomasse SED und dem sich ergebenden Sauerstoffdefizit (modelliert nach Sauerstoff-Isophleten realer Seen)

SED	Sauerstoff- defizit [mg/l]	SED	Sauerstoff- defizit [mg/l]
in 15 m Wassertiefe:			
0 .. 1	0,0	28 .. 29	4,5
2 .. 6	0,5	30	5,0
7 .. 10	1,0	31	5,5
11 .. 14	1,5	32	6,0
15 .. 17	2,0	33	6,5
18 .. 20	2,5	34	7,0
21 .. 23	3,0	35	7,5
24 .. 25	3,5	36 .. 43	28,0 - SED
26 .. 27	4,0	44 .. 49	28,0 - (2 * SED - 43)
		> 49	28,0 - (5 * SED - 190)
in 20 m Wassertiefe:			
0 .. 1	0,0	25 .. 26	6,5
2 .. 4	1,0	27	7,0
5 .. 6	1,5	28	7,5
7 .. 8	2,0	29	8,0
9 .. 10	2,5	30	8,5
11 .. 12	3,0	31	9,0
13 .. 14	3,5	32	7,5
15 .. 16	4,0	33 .. 39	23,0 - SED
17 .. 18	4,5	40 .. 46	23,0 - (2 * SED - 39)
19 .. 20	5,0	> 47	Defizit: 100%
21 .. 22	5,5		
23 .. 24	6,0		
in 25 m Wassertiefe:			
0 .. 1	0,0	3 .. 28	0,0 - (SED - 1)
2	2,0	30	0,0 - (2 * SED - 29)
		> 30	Defizit: 100%

Erläuterung der Abkürzungen

a,b	Halbachsen der elliptischen Oberfläche des Modellsees [m]
A	Fläche des Sees [m ²]
CUMSED	am Gewässergrund sedimentierende Biomassen (Hilfsvariable) [d ⁻¹]
E _i	Belastungseinheit der i-ten phosphatbelasteten Quelle
EPSAQUA	Extinktion durch das reine Wasser und gelöste Substanzen [dimensionslos]
ε	Extinktionskoeffizient für Licht (400 ... 700 nm) [dimensionslos]
F _{diff}	Phosphatfreisetzung aus dem Sediment [mg P · m ⁻² · d ⁻¹]
FTE	Faktor für Trübstoffeintrag [dimensionslos]
FTEMAX	Wert von FTE, bei dessen Überschreitung eine Abnahme der Lichttransmission durch Trübstoffeintrag einsetzt [dimensionslos]
FVZ	Nummer der Startdekade für Frühjahrsvollzirkulation
HVZ	Nummer der Startdekade für Herbstvollzirkulation
I	Licht (400 ... 700 nm) über der Wasseroberfläche [J · cm ⁻² · d ⁻¹]
I ₀	durch Reflexion reduziertes Licht (400 ... 700 nm) direkt unter der Wasseroberfläche [J · cm ⁻² · d ⁻¹]
I _{0z}	Licht (400 ... 700 nm) in der durchmischten Schicht [J · cm ⁻² · d ⁻¹]
KI	Halbsättigungskonstante für die Beziehung zwischen Licht (400 ... 700 nm) und Photosyntheserate [J · cm ⁻² · d ⁻¹]
k _I	Klimafaktor; Abhängigkeit der Globalstrahlung I von den zur Wahl stehenden klimatischen Bedingungen
KGZX	Halbsättigungskonstante für die Beziehung zwischen Grazing-Rate und Phytoplanktonbiomasse [cm ³ /m ³]
KX	Halbsättigungskonstante für die inverse Beziehung zwischen Photosyntheserate und Biomasse des Phytoplanktons [dimensionslos]
KXNUP	Phosphatausnutzungskoeffizient für Phytoplankton [cm ³ /mg P]
KXP	Halbsättigungskonstante für die Beziehung zwischen Phosphat und Photosyntheserate bei minimaler Phytoplanktonbiomasse [mg P/m ³]
KXP1	Konstante der Abhängigkeit des Halbsättigungswertes KX von der Phosphatkonzentration [cm ³ /mg P]
KXP2	Konstante der Abhängigkeit des Halbsättigungswertes KX von der Phosphatkonzentration [dimensionslos]
KZ	Halbsättigungswert für die inverse Beziehung zwischen Grazing-Rate und Biomasse des Zooplanktons [dimensionslos]
KZMO	Halbsättigungskonstante für die Beziehung zwischen Zooplanktonbiomasse und Mortalitätsrate des Zooplanktons [cm ³ /m ³]
KZNUP	Phosphatausnutzungskoeffizient für Zooplankton [cm ³ /mg P]
KZP1	Konstante der Abhängigkeit des Halbsättigungswertes KZ von der Phytoplanktonbiomasse [cm ³ /cm ³]
KZP2	Konstante der Abhängigkeit des Halbsättigungswertes KZ von der Phytoplanktonbiomasse [dimensionslos]
LTE	Logarithmus der prozentualen Lichttransmission bei 5 m Schichtdicke (Epilimnion) [dimensionslos]
p	Transformationsparameter zur näherungsweisen Ermittlung des Volumens des Wasserkörpers (extremes elliptisches Sinusoid) bei jeder gewünschten Wassertiefe ZMIXD [dimensionslos]
P ₁	Phosphatkonzentration im obersten Zentimeter des Interstitialwassers [mg P/l]
P _i	Fracht der i-ten phosphatbelasteten Quelle [g P/d]
PD	Orthophosphatkonzentration in der durchmischten Schicht [mg P/m ³]
PDEXP	Phosphatexport für durchmischte Schicht [mg P/d]
PDIMP	Phosphatimport für durchmischte Schicht [mg P/d]
PDKONS	Phosphatverbrauch durch Phytoplanktonwachstum in der durchmischten Schicht [mg P/d]
PDREMIN	Phosphatremineralisation in der durchmischten Schicht [mg P/d]
PEKKR	Rate der Phosphatremineralisation durch Exkretion des Zooplanktons [mg P · cm ⁻³ · d ⁻¹]

PH	Orthophosphatkonzentration im Hypolimnion [mg P/m^3]
PHOTPER	Lichtstunden pro Tag [h]
PHVZ	Phosphatkonzentration zu Beginn der Herbstvollzirkulation [mg P/m^3]
PIN	Orthophosphatkonzentration im Zufluß [mg P/m^3]
PZMORT	Rate der Phosphatremineralisation aus abgestorbenen oder von Fischen gefressenem Zooplankton [$\text{mg P} \cdot \text{cm}^{-3} \cdot \text{d}^{-1}$]
QDIN	Wasserzufluß in die durchmischte Schicht [m^3/d]
QDOUT	Wasserabfluß aus der durchmischten Schicht [m^3/d]
RI	Reflexionsfaktor für Licht [dimensionslos]
SBK ₁	spezifischer Belastungsquotient: Phosphatimport aus Niederschlägen $\text{SBK}_1 = 2,5 \text{ [g P/d] pro Einwohner}$
SBK ₂	spezifischer Belastungsquotient: Phosphatimport durch Abwässer aus Haushalten bei Verwendung phosphathaltiger Waschmittel $\text{SBK}_2 = 2,3 \text{ [g P/d] pro Einwohner}$
SBK ₃	spezifischer Belastungsquotient: Phosphatimport durch Abwässer aus Haushalten ohne Verwendung phosphathaltiger Waschmittel $\text{SBK}_3 = 1,7 \text{ [g P/d] pro Einwohner}$
SBK ₄	spezifischer Belastungsquotient: Phosphatimport durch Auswaschung landwirtschaftlicher Flächen $\text{SBK}_4 = 3,8 \text{ [g P/d] pro Hektar}$
SED	ganzzahliger Wert von CUMSED [dimensionslos]
SST	Nummer der Startdekade für Sommerstagnation
t	Zeit
TEMP	mittlere Wassertemperatur in ca. 5 m Tiefe [$^{\circ}\text{C}$]
TK1, TK2	Temperaturfaktor; Abhängigkeit der Wassertemperatur TEMP von den zur Wahl stehenden klimatischen Bedingungen [dimensionlos]
T _w	theoretische Wasseraustauschzeit [d]
UXZ	Physiologische Verwertbarkeit des Phytoplanktons durch das Zooplankton [dimensionslos]
V	Volumen des gesamten Wasserkörpers [m^3]
V _{z'}	Volumen des Wasserkörpers in der normierten Tiefe z' [dimensionslos]
VOLD	Volumen der durchmischten Schicht [m^3]
VOLH	Volumen des Hypolimnions [m^3]
X	Phytoplanktonbiomasse (-dichte) [cm^3/m^3]
XBPR	Brutto-Photosyntheserate des Phytoplanktons in der durchmischten Schicht [d^{-1}]
XBPRMAX	maximale Bruttphotosyntheserate des Phytoplanktons [d^{-1}]
XBPRMIN	Bruttphotosyntheserate bei Temperatur nahe 0°C und Optimalbedingungen bezüglich Licht und Phosphat [d^{-1}]
XEXT	Konstante der Lichtextinktion durch das Phytoplankton [m^3/cm^3]
XG	Phytoplanktonwachstum [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]
XGR	Wachstumsrate in der durchmischten Schicht (Nettoproduktion) [d^{-1}]
XGTOPT	Optimaltemperatur für Phytoplanktonwachstum [$^{\circ}\text{C}$]
XRESP	Respirationsrate des Phytoplanktons in der durchmischten Schicht [d^{-1}]
XRPP	Grundumsatz der Phytoplanktonrespiration [d^{-1}]
XS	Nettosinkgeschwindigkeit des Phytoplanktons [m/d]
XSEDI	Phytoplanktonverluste durch Sedimentation [d^{-1}]
XSEDRA	Sedimentationsrate des Phytoplanktons [d^{-1}]
WST	Nummer der Startdekade für Winterstagnation
z	mittlere Tiefe [m]
z'	normierte Tiefe [dimensionslos]
z _m	maximale Tiefe [m]
z _r	relative Tiefe [m]
Z	Zooplanktonbiomasse (-dichte) [cm^3/m^3]
ZAK	Verhältnis der assimilierten zur gefressenen Nahrungsmenge des Zooplanktons [cm^3/cm^3]

ZAKMAX	maximaler Assimilationskoeffizient bei sehr kleiner Grazing-Rate [cm^3/cm^3]
ZAKMIN	minimaler Assimilationskoeffizient bei $\text{ZGZRA} = \text{ZGZMAX}$ [cm^3/cm^3]
ZASSIM	Assimilationsrate des Zooplanktons [d^{-1}]
ZG	Zooplanktonwachstum [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]
ZGR	Wachstumsrate des Phytoplanktons [d^{-1}]
ZGL	Phytoplanktonverluste durch Freßtätigkeit (Grazing) des Zooplanktons [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]
ZGZDU	Grazing-Rate bei Dunkelheit [d^{-1}]
ZGLI	Lichthemmungsfaktor für Grazing [dimensionslos]
ZGLLI	Grazing-Rate bei Licht [d^{-1}]
ZGZMAX	maximale Grazing-Rate [d^{-1}]
ZGZRA	Grazing-Rate des Zooplanktons [d^{-1}]
ZGZTOPT	Optimaltemperatur für Grazing [$^{\circ}\text{C}$]
ZMEXD	Tiefe der durchmischten Schicht [m]
ZMORA	Mortalitätsrate des Zooplanktons [d^{-1}]
ZMORT	Zooplanktonmortalität [$\text{cm}^3 \cdot \text{m}^{-3} \cdot \text{d}^{-1}$]
ZMOMIN	Zooplanktonmortalitätsrate nahe 0°C bei großer Zooplanktonbiomasse ($Z \gg \text{KZMO}$) [d^{-1}]
ZMOT	Konstante der Temperaturabhängigkeit der Mortalitätsrate des Zooplanktons [$(^{\circ}\text{C} \cdot \text{d})^{-1}$]
ZRESP	Respirationsrate des Zooplanktons [d^{-1}]
ZRESPMIN	Respirationsrate des Zooplanktons bei Optimaltemperatur und fehlendem Nahrungsangebot [d^{-1}]
ZRESPOPT	Respirationsrate des Zooplanktons bei Optimaltemperatur und maximaler Grazing-Rate [d^{-1}]
ZRSPTMIN	Respirationsrate des Zooplanktons bei Temperatur nahe 0°C und optimalem Nahrungsangebot [d^{-1}]

COMPUTER-MODELL L A G O

1. War Ihr Vorwissen für die Arbeit mit dem Computer-Modell ausreichend ?

☐

ja

☐

nein

2. Schwierigkeitsgrad ? (Bitte 1 Aussage ankreuzen !)

☐

überfordernd

☐

schwer

☐

gerade richtig

☐

leicht

☐

zu leicht

3. Welche Vorteile bzw. Nachteile bietet Ihrer Meinung nach die Arbeit mit dem Computer-Modell ?

4. Wurden Inhalte vermittelt, die andere Lehrmedien nicht vermitteln können ?

☐

nein

☐

ja

Wenn ja: Welche ?

FORTSETZUNG: RÜCKSEITE

5. Wurden durch das Computer-Modell Zusammenhänge im Ökosystem See besonders deutlich ?

☐ nein

☐ ja

Wenn ja: Welche ?

6. Ich fand die Arbeit mit dem Computer-Modell

ermüdend	+++0+++	anregend
abwechslungsreich	+++0+++	monoton
anstrengend	+++0+++	einfach
interessant	+++0+++	langweilig
abstrakt	+++0+++	konkret
übersichtlich	+++0+++	unübersichtlich
anschaulich	+++0+++	unanschaulich
folgerichtig	+++0+++	zusammenhanglos

Anlage 5

M E N Ü S

(Listen auf dem Bildschirm zur Auswahl von Parametern)

SIMULATION: NORMALLAUF 1

SIMULATION: EXPERIMENTE 2
(Es können einige zusätzliche
Parameter verändert werden.)

IHRE AUSWAHL: 

Menü 1: Wahl des Simulationslaufes

Folgende PARAMETER können zu Experimentierzwecken verändert werden:

Optimaltemperatur	
für Algenwachstum [°C]	: 20.0
für Zooplanktonfraß [°C]	: 20.0
Phosphatausnutzungskoeffizient	
für Algen [cm3/mg P]	: 0.8
für Zooplankton [cm3/mg P] ..	: 0.8
Faktor für die physiologische	
Verwertbarkeit der Algen	
durch das Zooplankton	: 1.0
Lichthemmungsfaktor	
für Zooplanktonfraß	: 0.3
Nettosinkgeschwindigkeit	
der Algen [m/d]	: 0.1

Simulation fortsetzen	1
Daten ändern	2

Ihre Eingabe: ■

Menü 2: Veränderung zusätzlicher Parameter für Experimente mit dem Computerelement

ANFANGSZUSTAND DES SEES zu Beginn der Simulation:
(zur Auswahl)

- nährstoffarm 1
(oligotroph)
- leicht nährstoffreich 2
(mesotroph)
- nährstoffreich 3
(eutroph)

Ihre Auswahl: ☒

Menü 3: Wahl des trophischen Zustandes des Modellsees

KLIMATISCHE BEDINGUNGEN (zur Auswahl)

- warmer Sommer,
warmer Winter,
wenig Bewölkung 1
- normaler Sommer,
normaler Winter 2
- kühler Sommer,
kalter Winter,
häufige Bewölkung 3

Ihre Auswahl: ☐

Menü 4: Wahl der Klimatischen Bedingungen

P H O S P H A T - I M P O R T *

158

=====

aus der Gemeinde SEEDORF

EINWOHNERZAHL der Gemeinde SEEDORF ... : 100
 Prozentsatz der Einwohner, die
 phosphathaltige Düngemittel benutzen . : 100 %
 direkt an den See grenzende
 gedüngte Wiesen- und Ackerflächen : 100 ha

Behandlung des ABWASSERS nach Situation: A

zur Auswahl:

Die Abwässer werden in den See geleitet:

A: direkt

über eine KLÄRANLAGE mit der(n) Reinigungsstufe(n)

B: mechanisch

C: mechanisch und biologisch

D: mechanisch, biologisch und chemisch

B I L A N Z **

=====

atmosphärische Fracht (aus Niederschlägen)	12.0 g P/Tag
Abwässer aus den Haushalten	690.0 g P/Tag
Auswaschung landwirtschaftlicher Flächen	760.0 g P/Tag

Summe : 1462.0 g P/Tag

Bei einer Zuflußmenge von 33704 m³/Tag ergibt sich

eine Ortho-Phosphatkonzentration von : 43.4 mg P/Tag

Simulation fortsetzen	1
Daten ändern	2
Berechnungsgrundlagen einsehen ..	3

Ihre Eingabe: █

* nur gelöstes Ortho-Phosphat ** Jahresdurchschnittswerte

Menü 5: Festlegung der Größe des Phosphatimports

BERECHNUNGSGRUNDLAGEN =====

Phosphat-Import aus Niederschlägen : 12 g P / Tag (Festwert)

Phosphat-Import durch Abwässer aus Haushalten :

- bei Verwendung phosphathaltiger Waschmittel : 2,3 g P / Tag pro Einwohner
- ohne Verwendung phosphathaltiger Waschmittel: 1,7 g P / Tag pro Einwohner

Phosphat-Import durch Auswaschung
landwirtschaftlicher Flächen : 3,8 g P / Tag pro Hektar

Nur Kläranlagen mit chemischer Reinigungsstufe vermindern die Phosphate
(um ca. 90 %).

zurück zur Simulation:
beliebige Taste drücken !

Menü 6: Berechnungsgrundlagen für Phosphatimport

SIMULATION FORTSETZEN 1
(nächstes Modelljahr)

SIMULATION WIEDERHOLEN 2
gleiches Modelljahr,
andere Parameter)

SIMULATION BEENDEN 3

IHRE AUSWAHL: ☐

Menü 7: Wahl der Fortsetzung der Simulation

A U S D R U C K E

(Hardcopies)

Die in den Ausdrucken dokumentierten Modellläufe wurden mit folgenden *PARAMETEREINSTELLUNGEN* durchgeführt:

alle Ausdrücke: *NORMALLAUF*

Ausdruck 1 - 7 (1. Modelljahr): Anfangszustand des Sees: *nährstoffarm*; klimatische Bedingungen: *normaler Sommer/normaler Winter*;
Phosphatimport *43,4 mg/m³*

Ausdruck 8 (1. Modelljahr): Anfangszustand des Sees: *nährstoffarm*; klimatische Bedingungen: *warmer Sommer/warmer Winter/wenig Bewölkung*;
Phosphatimport *43,4 mg/m³*

Ausdruck 9 (1. Modelljahr): Anfangszustand des Sees: *nährstoffarm*; klimatische Bedingungen: *kühler Sommer/kalter Winter/häufige Bewölkung*;
Phosphatimport *43,4 mg/m³*

Ausdruck 10 - 11 (1. Modelljahr): Anfangszustand des Sees: *nährstoffarm*; klimatische Bedingungen: *normaler Sommer/normaler Winter*;
Phosphatimport *6,3 mg/m³*

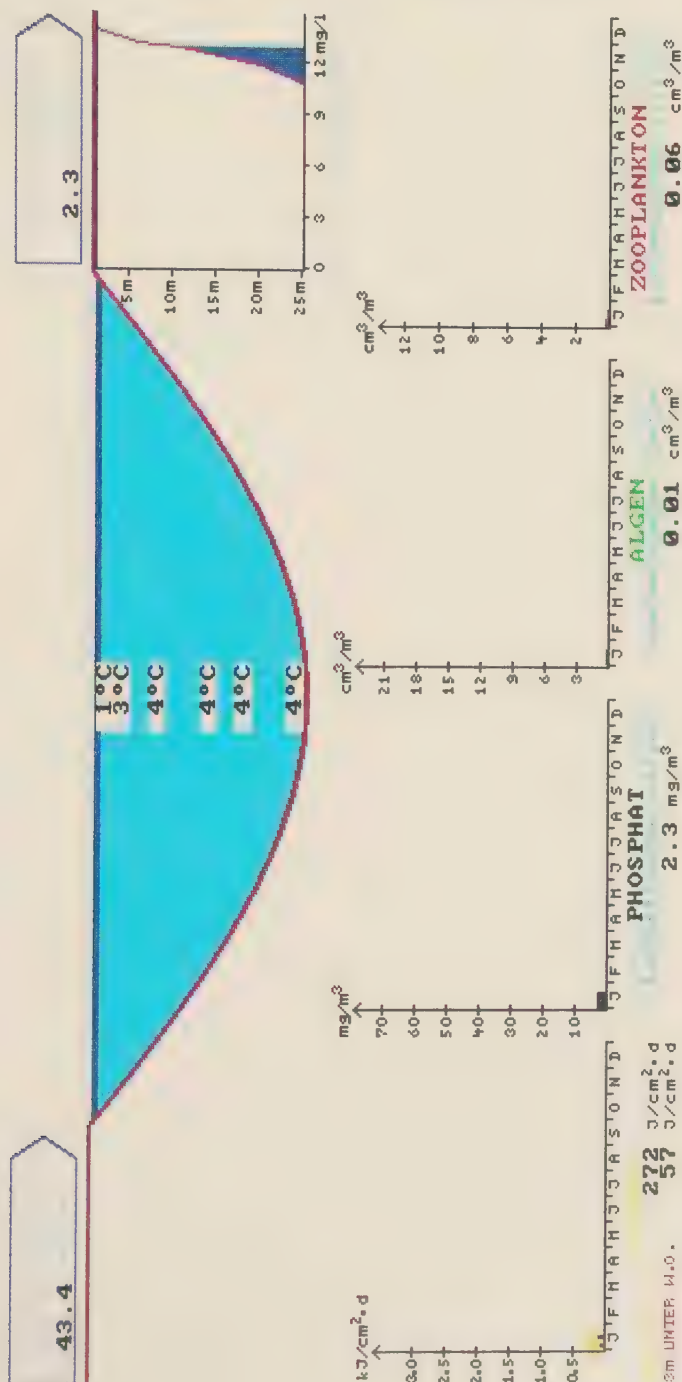
Ausdruck 12 - 14 (3. Modelljahr): klimatische Bedingungen: *normaler Sommer/normaler Winter*;
Phosphatimport *43,4 mg/m³*

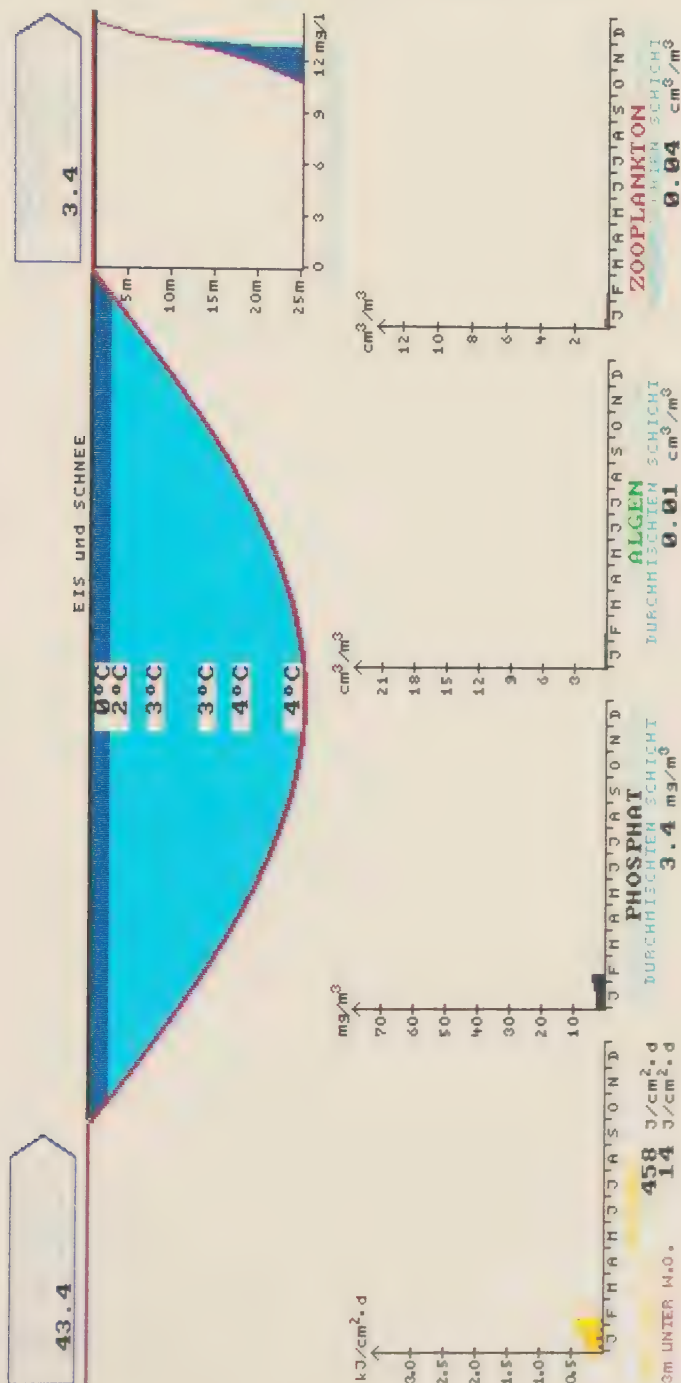
Ausdruck 15 - 16 (3. Modelljahr): klimatische Bedingungen: *warmer Sommer/warmer Winter/wenig Bewölkung*;
Phosphatimport *43,4 mg/m³*

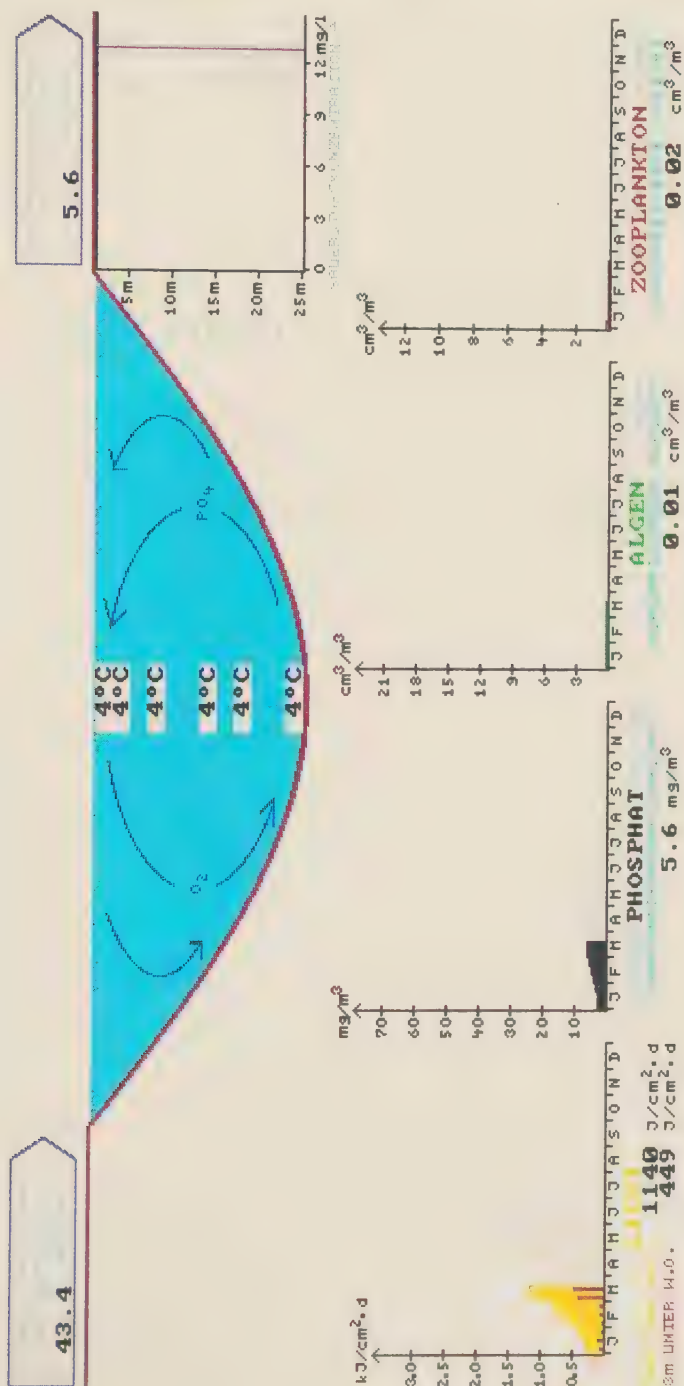
ab 4. Modelljahr: Phosphatimport *75,8 mg/m³*

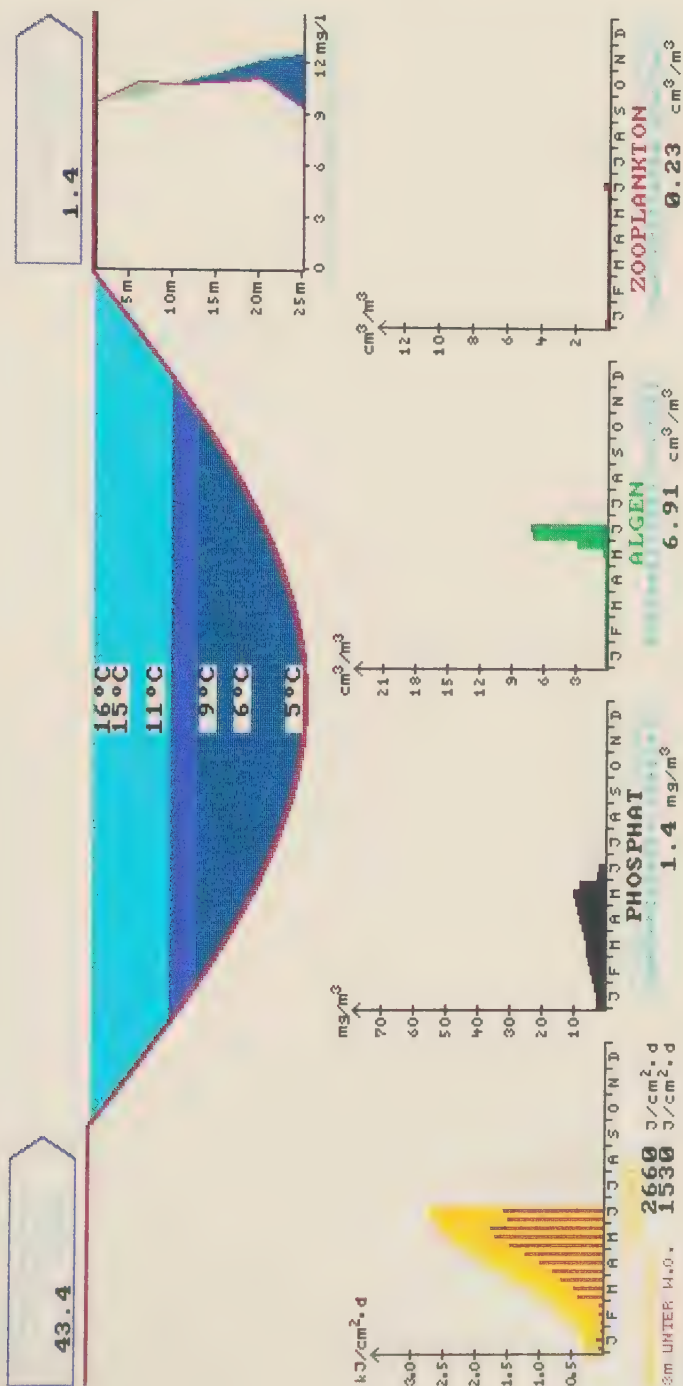
Ausdruck 17 (5. Modelljahr): klimatische Bedingungen: *warmer Sommer/warmer Winter/wenig Bewölkung*;
Phosphatimport *75,8 mg/m³*

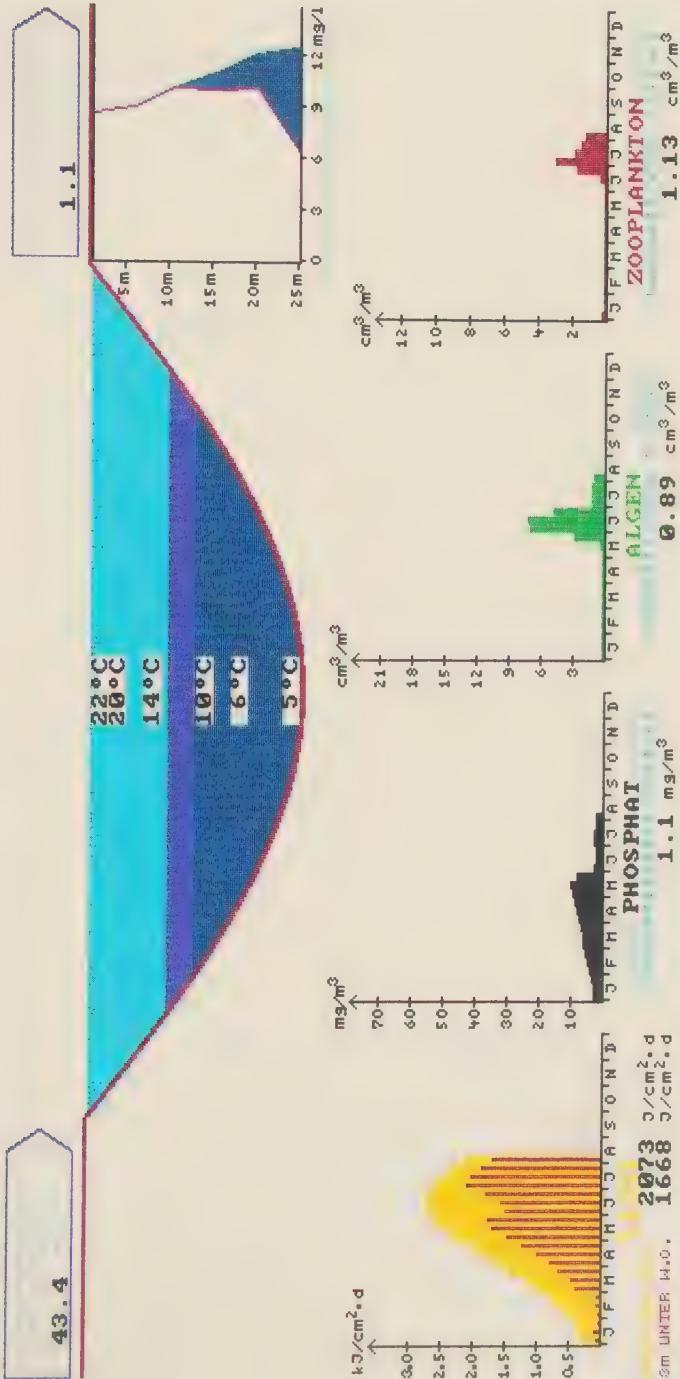
Ausdruck 18 (6. Modelljahr): klimatische Bedingungen: *warmer Sommer/warmer Winter/wenig Bewölkung*;
Phosphatimport *75,8 mg/m³*

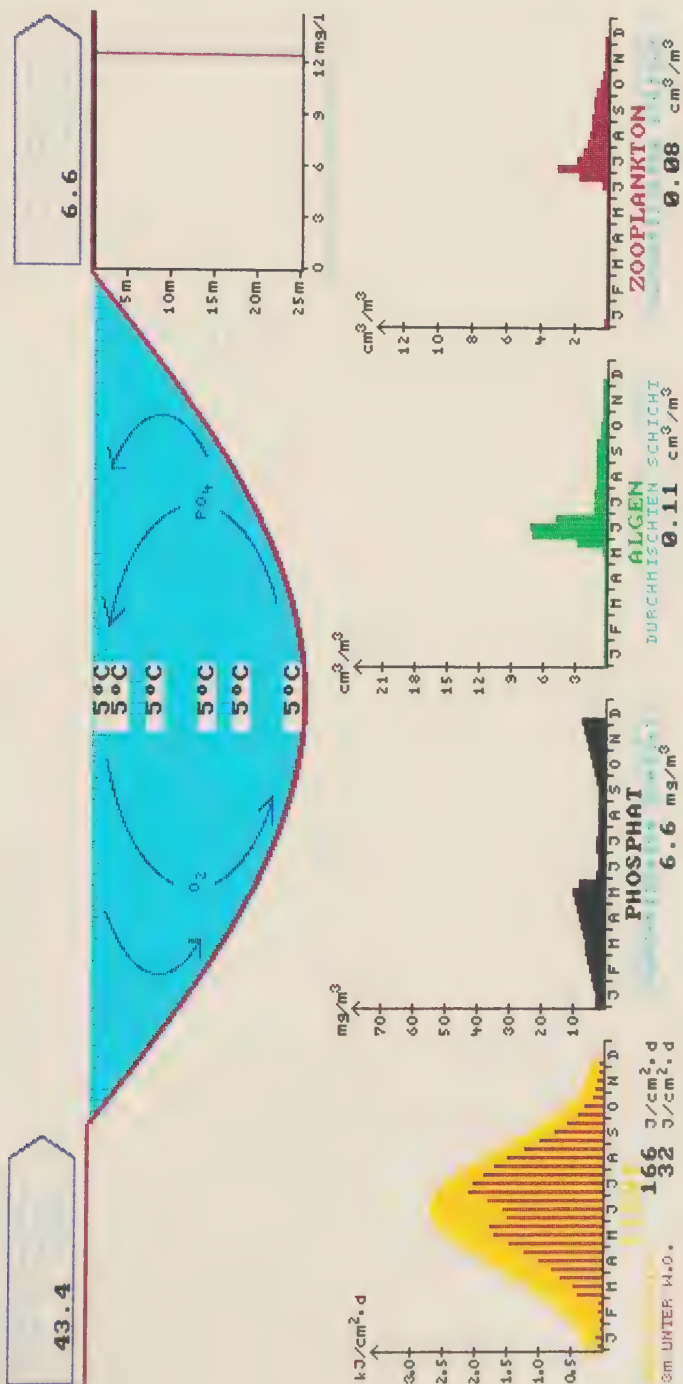


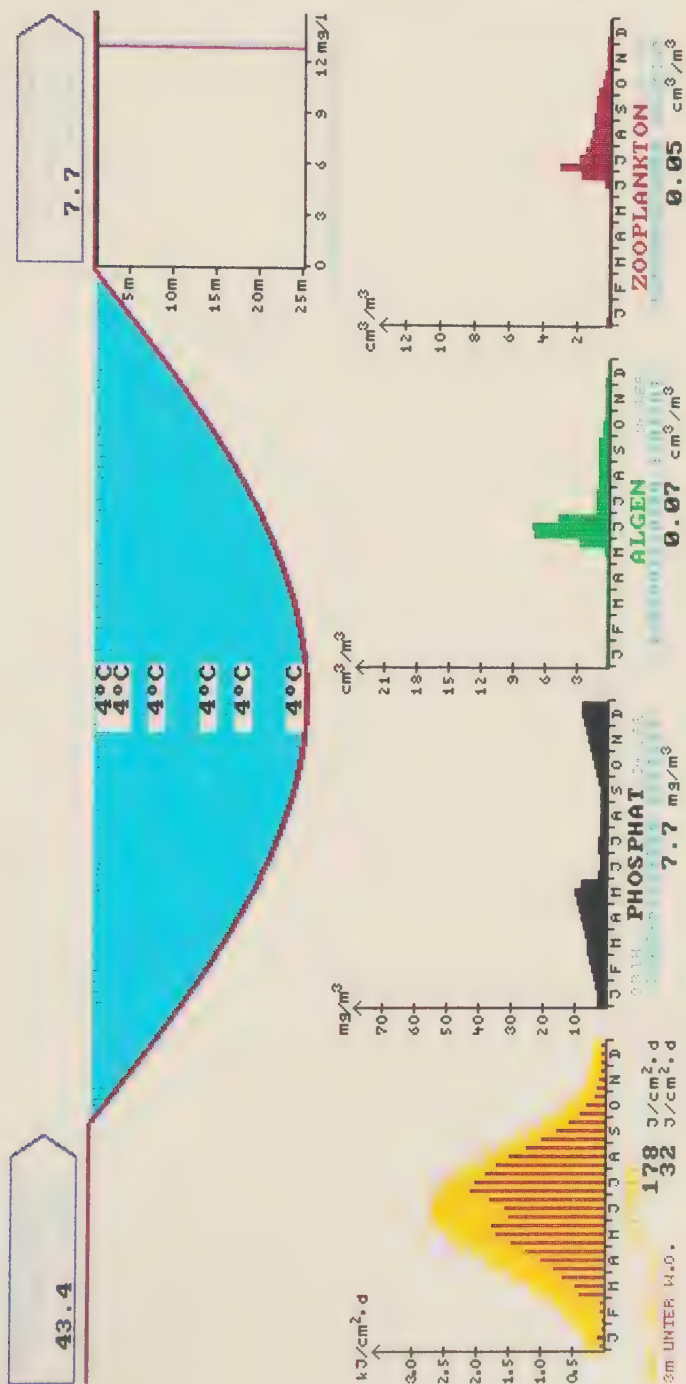


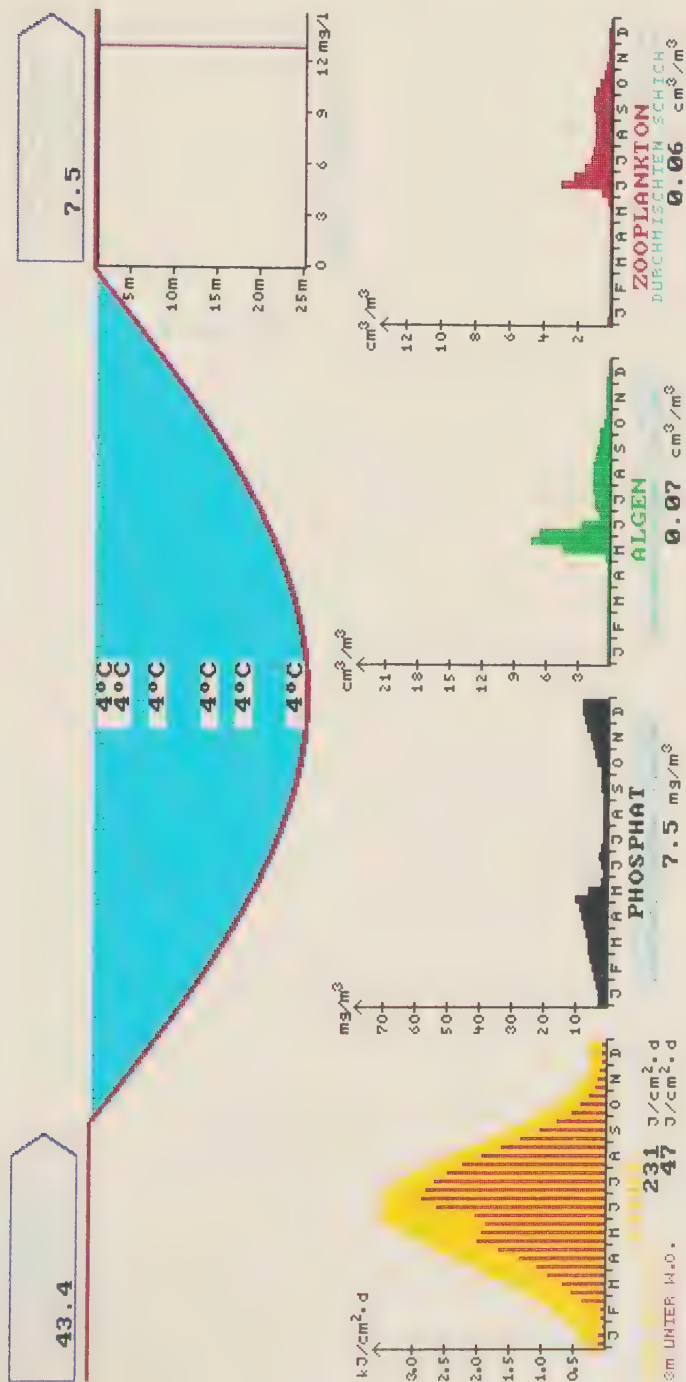


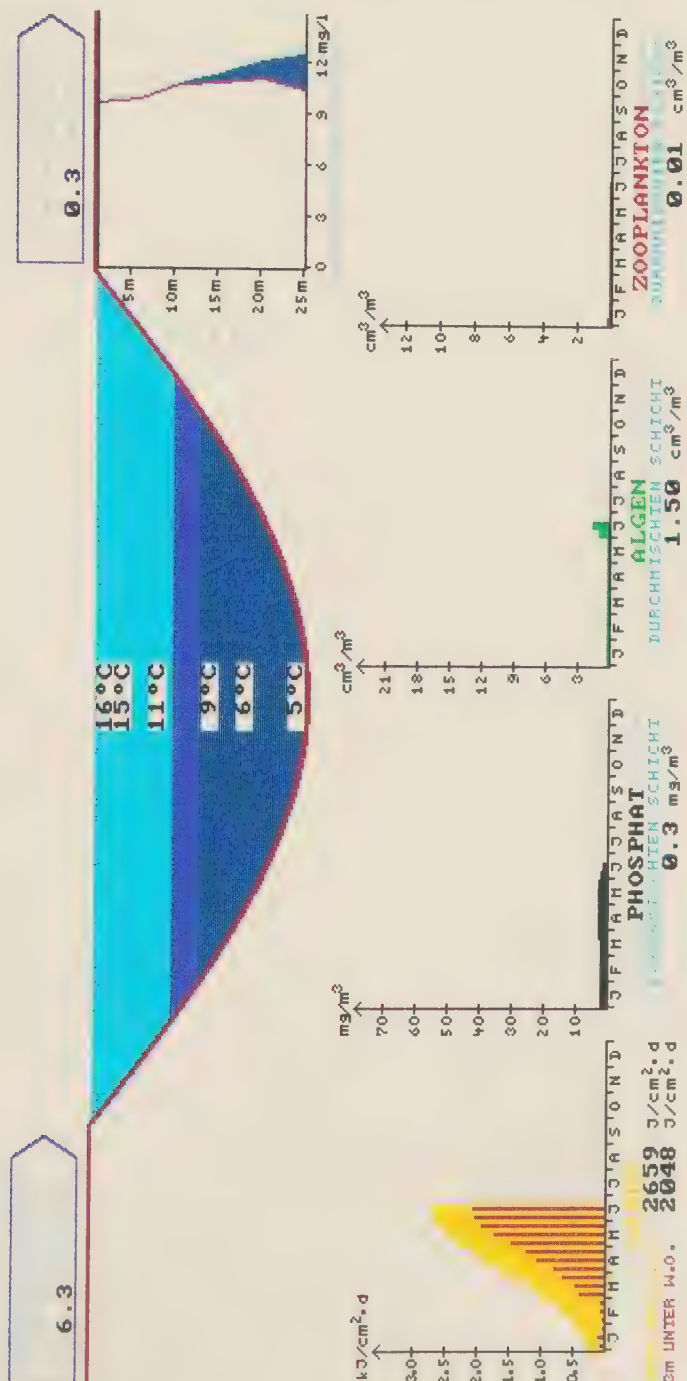


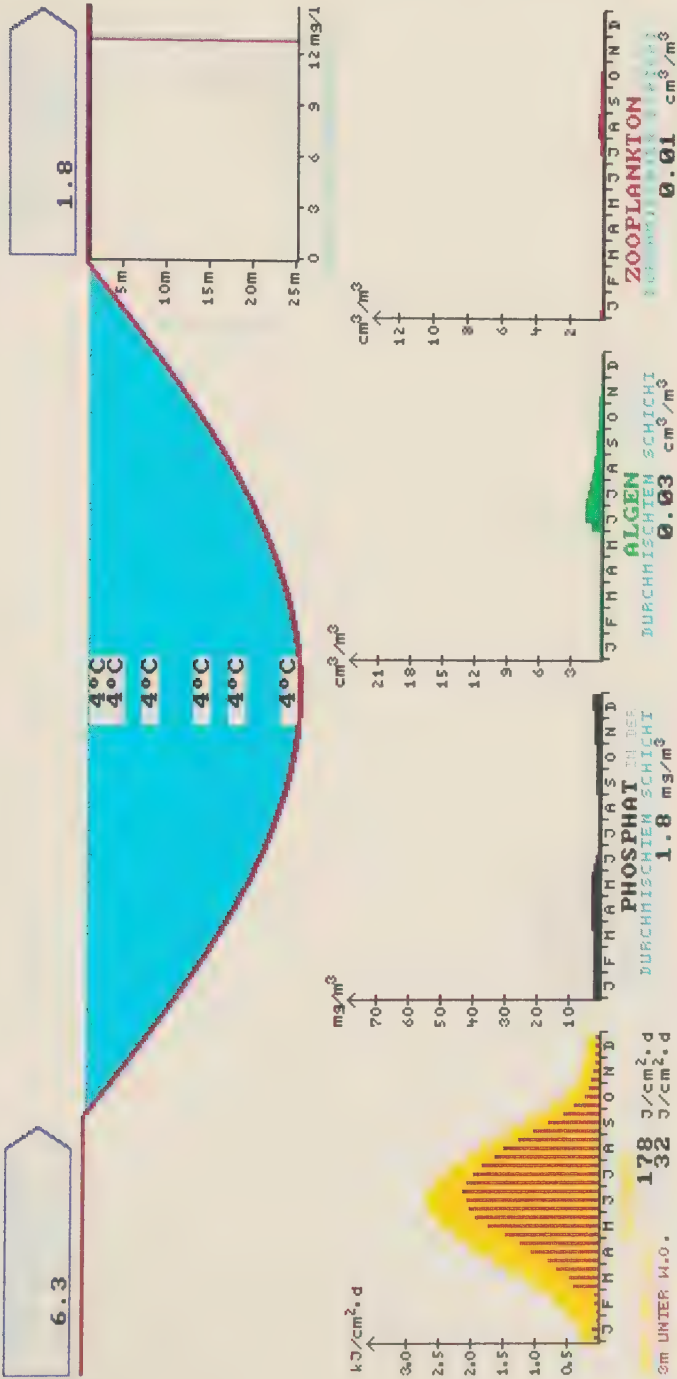


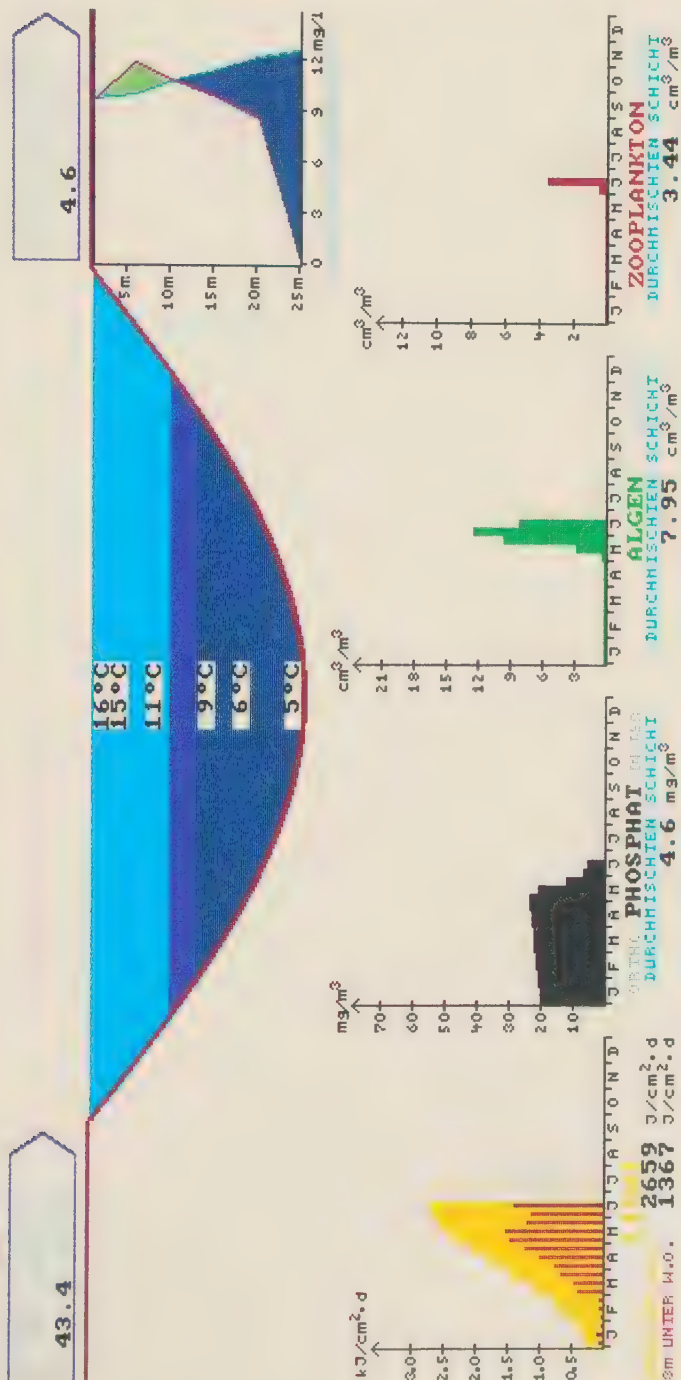












ORTHO-PHOSPHAT

SAUERSTOFFMANGEL

43.4

1.3

22°C
20°C
14°C

10°C
6°C
5°C

P.O.₄
P.O.₄

kJ/cm².d
3.0
2.5
2.0
1.5
1.0
0.5

ms/m³

cm³/m³

cm³/m³

2073 J/cm².d
1666 J/cm².d

PHOSPHAT

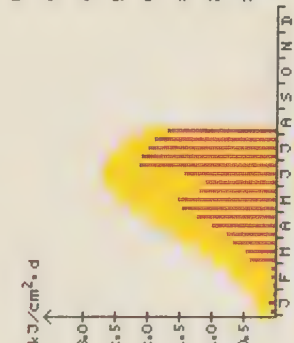
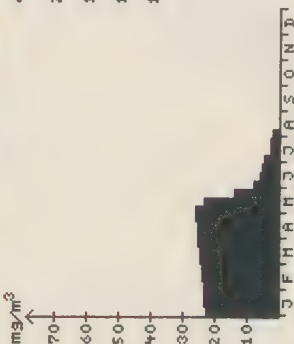
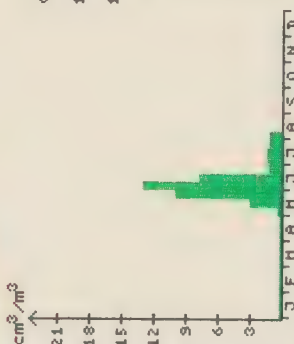
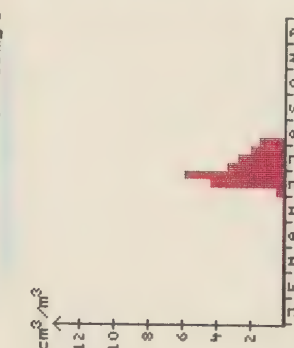
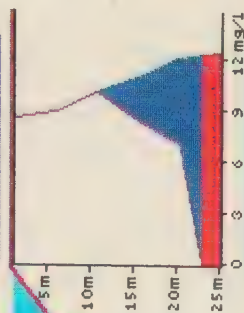
1.3 mg/m³

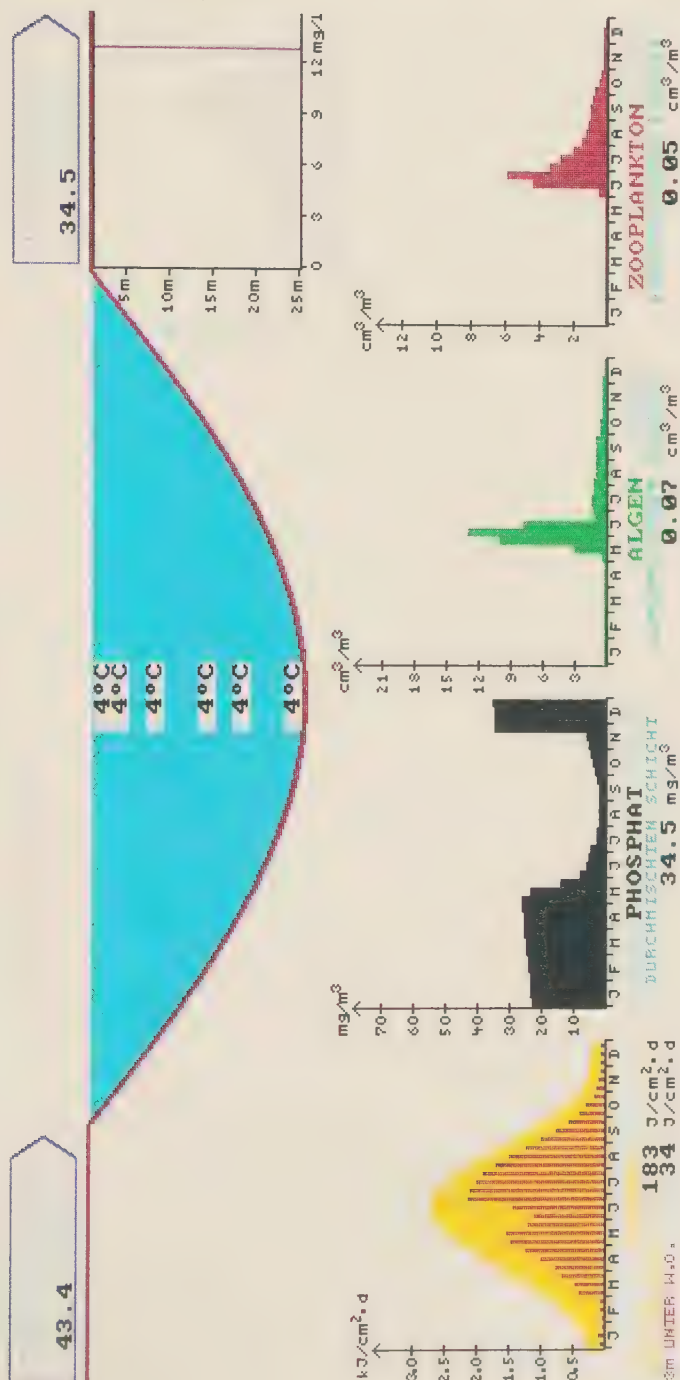
ALGEN

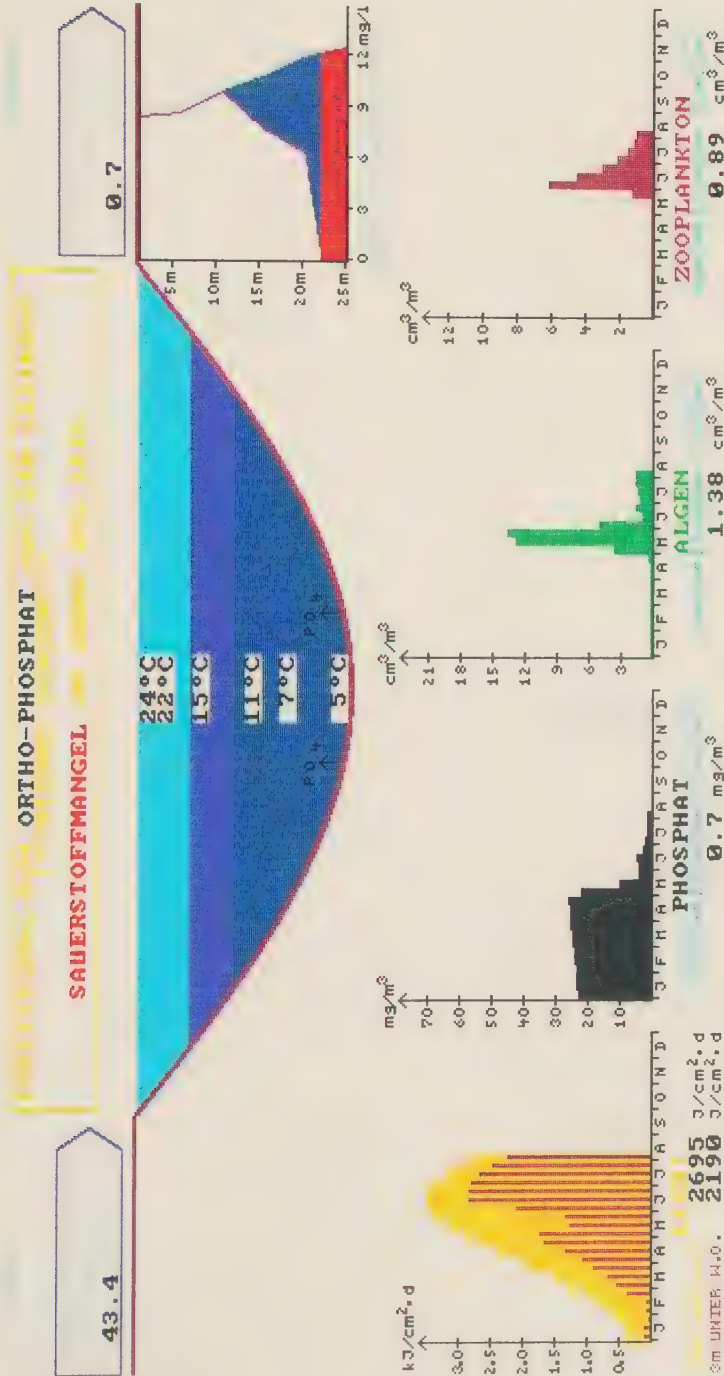
0.92 cm³/m³

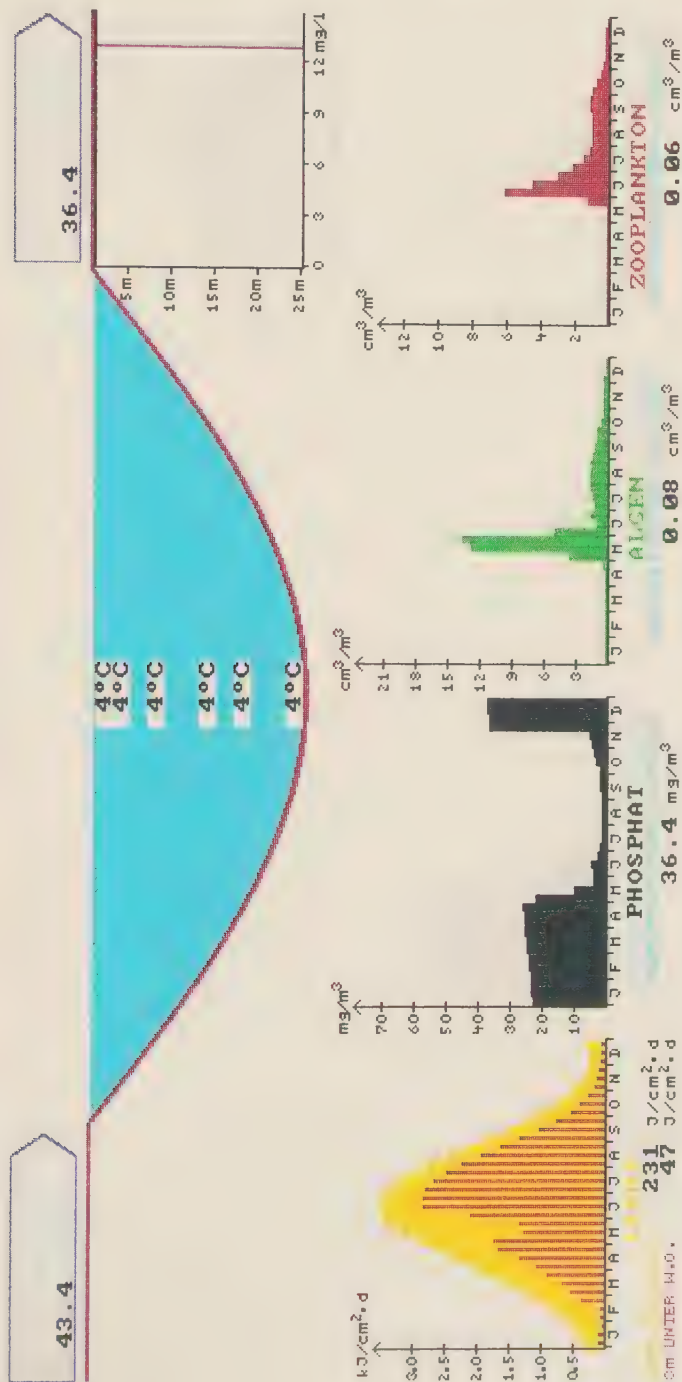
ZOOPLANKTON

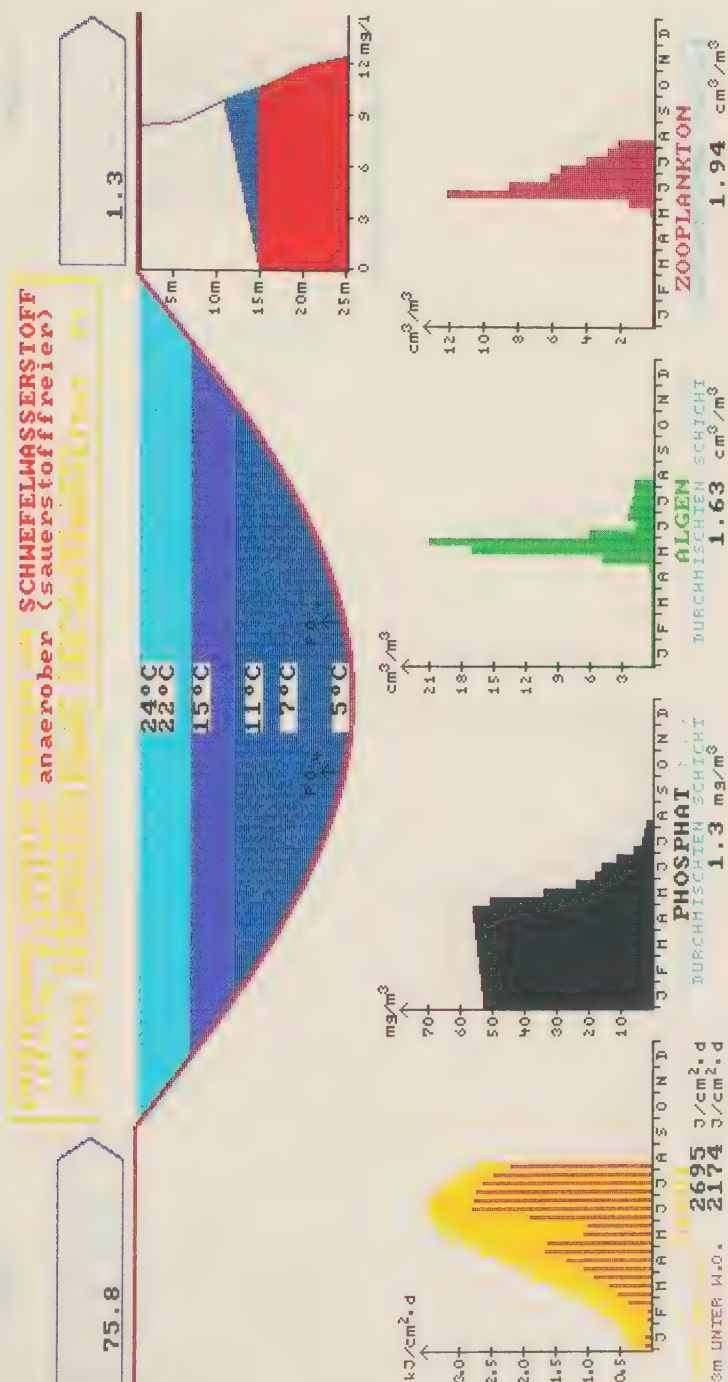
1.38 cm³/m³

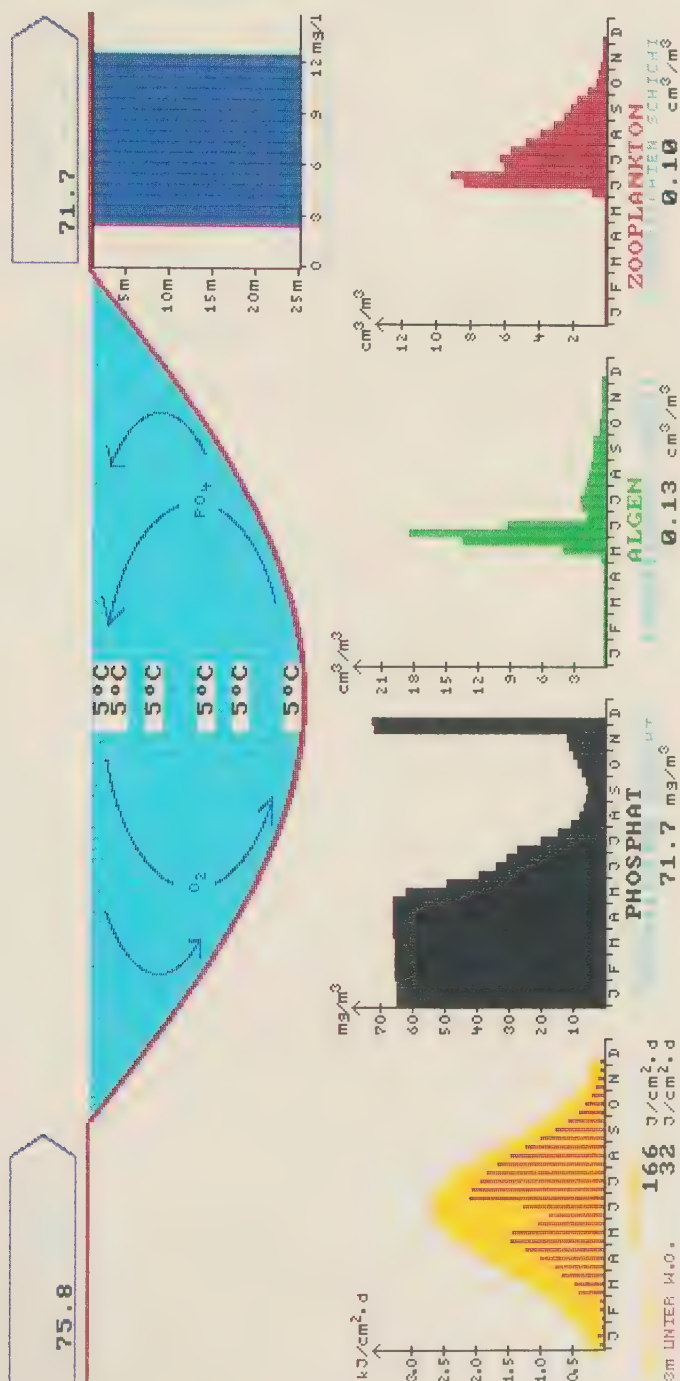












Anlage 7: Computerprogramm**Computerprogramm für das
Computermodell LAGO****Version 3 für Enhanced Graphics Adapter (EGA)****Programstruktur:****Hauptdatei:** LAGO.PAS**weitere Module:****INTRO.OVL
TSCHICHT.OVL
HINWEISE.OVL
LAGOPROC.INC
FONT4x6.INC
TSCHICHT.INC**

LAGO.PAS

HAUPTDATEI

```
program lago_EGA;
```

```
|
  Version für Enhanced Graphics Adapter (EGA)
```

```
  EGA-Modus E2/E3 :
```

```
  Grafik-Modus mit Auflösung 640x350 und 16 Farben aus einer Palette von 64
```

```
  Enhanced-Color-Bildschirm; Bildschirmspeicher 256 KByte;
```

```
  aktueller Stand: 01.10.1987
```

```
  Copyright: Peter G. Hiering, Carl-Orff-Weg 10, 8025 Unterhaching
```

```
|
  {Compilerbefehle}
```

```
  {SC-,U-,R-}
```

```
  {Konstantendefinition}
```

```
const  TANF = 0;           {Startzeit}
       L    = 2000;        {Laenge des Modellsees in m}
       ZMAX = 25;          {Tiefe des Modellsees in m}
       KI    = 29.0;       {Halbsaettigungskonstante fuer die Beziehung zwischen
                           Licht und Photosyntheserate}

       YAUN = 39;          {Abstand der x- bzw. y-Achse vom unteren Fensterrand}
       YAOB = 9;           {Abstand der y-Achse vom oberen Fensterrand}
       XALI = 14;          {Abstand der x-Achse vom linken Fensterrand}
       XARE = 0;           {Abstand der x-Achse vom rechten Fensterrand}
```

```
  {Typdefinition}
```

```
type   wko      = (xol,yol,xur,yur);           {Fensterkoordinaten}
       jz       = (j0,j5,j10,j15,j20,j25);    {Tiefenschichten}

       typerepack = record case integer of      {Register fuer DOS-Interrupt}
         1: (ax,bx,cx,dx,bp,si,di,ds,es,flags: integer);
         2: (al,ah,bl,bh,cl,ch,dl,dh:           byte);
       end;
```

{Vereinbarung der lokalen Variablen}

```

var    i,                {Laufvariable}
        t,                {Zeit in [Tagen]}
        h,                {Zeitschritt bei Runge-Kutta}
        zmixd,            {Durchmischungstiefe}
        versetz,          {Wert fuer horizontale Anordnung der Histogramm-
                           balken}
        ew,               {Zahl der Einwohner}
        pzsatz,           {Prozentsatz der Einwohner mit phosphath. Waschm.}
        lwfl,             {landwirtschaftliche Flaeche}
        sed,              {Integer-Wert von cumsed}
        wertoverxmax,     {Bereichsuebertrag Algen}
        wertoverzmax,     {Bereichsuebertrag Zooplankton}
        e_text_page,      {Textseite in EGA-Speicher}
        e_text_attribute  {Textattribute}
        :integer;

time,          {Zwischenspeicher fuer Zeit}
deka,          {Dekade}
jahr,          {Modelljahr}
klimasign,     {Flag fuer gewaehlte klimatische Bedingungen}
sst,           {Startdekade fuer Sommerstagnation}
hvv,           {Startdekade fuer Herbstvollzirkulation}
:byte;

entsorg        {Abwasserbehandlung}
:char;

monat          {Monate}
:string[9];

temp,          {Wassertemperatur}
photper,       {Tageslaenge}
li,            {Licht ueber der Gewaesseroberflaeche}
li0,           {durch Reflexion reduziertes Licht}
IOJ5m,         {Licht in 5 m Wassertiefe}
eps,           {Extinktionskoeffizient fuer Licht}
pd,            {Orthophosphatkonzentration in der durchmischten
               Schicht}
pdanf,         {Orthophosphatkonzentration in der durchmischten
               Schicht zu Jahresbeginn}
ph,            {Orthophosphatkonzentration im Hypolimnion}
pin,           {Orthophosphatkonzentration im Zufluß}
pdimp,         {Phosphatimport fuer durchmischte Schicht}
pdexp,         {Phosphatexport fuer durchmischte Schicht}
x,             {Phytoplanktonbiomasse}
xanf,          {Phytoplanktonbiomasse zu Jahresbeginn}
xg,            {Phytoplanktonwachstum}
xgr,           {Wachstumsrate des Phytoplanktons in der
               durchmischten Schicht}
xsedi,         {Phytoplanktonverluste durch Sedimentation}
z,             {Zooplanktonbiomasse}
zanf,          {Zooplanktonbiomasse zu Jahresbeginn}
zg,            {Zooplanktonwachstum}

```

```

zgr,           {Wachstumsrate des Zooplanktons}
zmort,         {Zooplanktonmortalitaet}
zg2,           {Phytoplanktonverluste durch grazing}
tk1,tk2,       {Koeffizienten fuer trig. Funktion Wassertemperatur}
lk,            {Koeffizient fuer trig. Funktion Globalstrahlung}
wf,wf2,        {Phase}
cumsed,        {sedimentierte Biomasse; kumulierter Wert}
cumsedanf,     {wie dto.; Wert zu Jahresbeginn}
vol,           {Volumen des Sees}
vold,          {Volumen der durchmischten Schicht}
vole_5m,vole_7m,
vole_12m,vole_16m,
vole_20m,      {Volumen des Wasserkoeperpers bis _xx m Tiefe}
sedarea,       {Flaeche der Sedimentschicht}
qdin,          {Wasserzufluß in der durchmischten Schicht}
xgtopt,        {Optimaltemperatur fuer Phytoplanktonwachstum}
zgztot,        {Optimaltemperatur fuer grazing}
kxnp,          {Phosphatausnutzungskoeffizient fuer Phytoplankton}
kznup,         {Phosphatausnutzungskoeffizient fuer Zooplankton}
uxz,           {Physiologische Verwertbarkeit des Phytoplanktons
               durch das Zooplankton}
zgzi,          {Lichthemmungsfaktor fuer grazing}
xs             {Nettosinkgeschwindigkeit des Phytoplanktons}
:real;

{diverse Flags}
drawflag, flag_intd,
flag_h2s, pfeiflag,
flag_abbruch,flag_last,
endstadium1,endstadium2,
endstadium3                                         :boolean;

{Arrays fuer Grafik Seebecken, Sauerstoffkurven, Durchmischungspfeile}
tj:           array[jz,1..36] of integer;
osat:         array[jz,1..36] of real;
odef:         array[jz] of real;
points_osat,points_oakt:
win(dow):     array[1..8,wko] of integer;
wly(waterline),basinline1,
basinline2,basinline3:
hxl,hxr:{hatch-coord.}
array[1..400] of integer;
array[0..100] of integer;
flashx1,flashy1:
array[255..390] of integer;
flashx2,flashy2:
array[450..555] of integer;
flashx3,flashy3:
array[315..495] of integer;
flashx4,flashy4:
array[699..804] of integer;
flashx5,flashy5:
array[864..999] of integer;
flashx6,flashy6:
array[756..936] of integer;

```

```

{$I egaproc.INC}
{$I font4x6.INC}
{$I tschicht.INC}
{$I tempvert.OVL}
{$I intro.OVL}
{$I hinweise.OVL}

{ ----- Tastaturpuffer loeschen ----- }
procedure flush_keyboard;
var regs: typeregpack;
begin
  regs.ax := $0C00;
  Msdos(regs);
end;

{ ----- Unterbrechung des Programmlaufs ermoeglichen ----- }
procedure pruefstop;
var stopchar: char; {Charakter fuer Pruefroutine}

begin
  if KeyPressed
  then begin
    read(Kbd,stopchar);
    stopchar := UpCase(stopchar);
    if stopchar = Chr(83) then repeat until keypressed;
    flush_keyboard;
  end;
end;

{ ----- blinkender Cursor fuer Menues ----- }
procedure blink_cursor(xpos,ypos,co:byte);
begin
  repeat
    textattr(co,0); write(' '); e_gotoxy(xpos,ypos,0);
    delay(200);
    textattr( 0,0); write(' '); e_gotoxy(xpos,ypos,0);
    delay(200);
  until keypressed;
end;

{ ----- Festlegung von Durchmischungstiefe, -----
      Volumen der durchmischten Schicht,
      Startdekade von SST und HVZ }
procedure durchmischungstiefe;
begin
  if klimasign = 1
  then begin
    sst := 13;
    hvz := 33;
    if ((deka >= 1) and (deka < 10)) or
       ((deka >= 34) and (deka <= 36)) then begin
      zmixd := ZMAX;
      vold := vol;
    end;
  end;
end;

```

```

if (deka = 10) or
  ((deka >= 32) and (deka < 34)) then begin
    zmixd := 16;
    vold := vole_16m;
  end;

if ((deka >= 11) and (deka < 13)) or
  ((deka >= 29) and (deka < 32)) then begin
    zmixd := 12;
    vold := vole_12m;
  end;

if (deka >= 13) and (deka < 29) then begin
    zmixd := 5;
    vold := vole_5m;
  end;

end

else
if klimasign = 2
then begin
  sst := 14;
  hvz := 33;
  if ((deka >= 1) and (deka < 10)) or
    ((deka >= 34) and (deka <= 36)) then begin
    zmixd := ZMAX;
    vold := vol;
  end;

  if ((deka >= 10) and (deka < 12)) or
    ((deka >= 30) and (deka < 34)) then begin
    zmixd := 16;
    vold := vole_16m;
  end;

  if ((deka >= 12) and (deka < 14)) or
    ((deka >= 28) and (deka < 30)) then begin
    zmixd := 12;
    vold := vole_12m;
  end;

  if ((deka >= 14) and (deka < 28)) then begin
    zmixd := 7;
    vold := vole_7m;
  end;

end

else
if klimasign = 3
then begin
  sst := 17;
  hvz := 34;
  if ((deka >= 1) and (deka < 12)) or
    ((deka >= 34) and (deka <= 36)) then begin
    zmixd := ZMAX;
    vold := vol;
  end;

  if ((deka >= 12) and (deka < 14)) or
    ((deka >= 32) and (deka < 34)) then begin
    zmixd := 20;
    vold := vole_20m;
  end;

```

```

if ((deka >= 14) and (deka < 17)) or
  ((deka >= 29) and (deka < 32)) then begin
    zmixd := 16;
    vold := vole_16m;
  end;
if ((deka >= 17) and (deka < 29)) then begin
    zmixd := 12;
    vold := vole_12m;
  end;
end;
end;

```

{ ————— numerische Approximation der Differentialgleichungen ————— }

procedure runge_kutta;

```

var j: byte; {Laufzeitvariable}
    k,l,m: array[1..4] of real; {Korrekturfaktoren}
    tr,dx,xr,dz,zr,dp,pr: real; {Variablen fuer die
                                   Ablage der Zwischen-
                                   ergebnisse}

```

procedure model; {lokal of runge_kutta}

const

```

KGZX = 5.0; {Halbsättigungskonstante für die Beziehung zwischen Grazing-Rate und
             Phytoplanktonbiomasse}
KXP = 1.7; {Halbsättigungskonstante für die Beziehung zwischen Phosphat und
            Photosyntheserate bei minimaler Phytoplanktonbiomasse}
KXP1 = 2.8; {Konstante der Abhängigkeit des Halbsättigungswertes KX von der
            Phosphatkonzentration}
KXP2 = 0.4; {Konstante der Abhängigkeit des Halbsättigungswertes KX von der
            Phosphatkonzentration}
KZMO = 0.35; {Halbsättigungskonstante für die Beziehung zwischen Zooplankton-
              biomasse und Mortalitätsrate des Zooplanktons}
KZP1 = 5.7; {Konstante der Abhängigkeit des Halbsättigungswertes KZ von der
            Phytoplanktonbiomasse}
KZP2 = 0.4; {Konstante der Abhängigkeit des Halbsättigungswertes KZ von der
            Phytoplanktonbiomasse}
XBPRMAX = 1.8; {maximale Bruttphotosyntheserate des Phytoplanktons}
XBPRMIN = 0.2; {Bruttphotosyntheserate bei Temperatur nahe 0°C und Optimal-
               bedingungen bezüglich Licht und Phosphat}
XEXT = 0.2; {Konstante der Lichtextinktion durch das Phytoplankton}
ZAKMAX = 0.8; {maximaler Assimilationskoeffizient bei sehr geringer Grazing-Rate}
ZAKMIN = 0.4; {minimaler Assimilationskoeffizient bei ZGZRA=ZGZMAX}
ZGZMAX = 2.5; {maximale Grazing-Rate}
ZMOMIN = 0.01; {Zooplanktonmortalitätsrate nahe 0°C bei großer Zooplanktonbiomasse}
ZMOT = 0.004; {Konstante der Temperaturabhängigkeit der Mortalitätsrate des
               Zooplanktons}
ZRESPMIN = 0.1; {Respirationsrate des Zooplanktons bei Optimaltemperatur und
                 fehlendem Nahrungsangebot}
ZRESPOPT = 0.2; {Respirationsrate des Zooplanktons bei Optimaltemperatur und
                 maximaler Grazing-Rate}
ZRSPTMIN = 0.05; {Respirationsrate des Zooplanktons bei Temperatur nahe 0°C und
                  optimalem Nahrungsangebot}

```

```

var lte,      {Logarithmus der prozentualen Lichttransmission bei 5 m Schichtdicke}
    IQJ,      {Licht(400...700nm) in der durchmischten Schicht}
    kx,        {Halbsättigungswert für die inverse Beziehung zwischen Photosyntheserate und
                Biomasse des Phytoplanktons}
    kz,        {Halbsättigungswert für die inverse Beziehung zwischen Grazing-Rate und Bio-
                masse des Zooplanktons}
    xbpr,      {Brutto-Photosyntheserate des Phytoplanktons in der durchmischten Schicht}
    xbprjt1,   {ausgeklammerter Teil zur Errechnung von xbpr}
    xresp,     {Respirationsrate des Phytoplanktons in der durchmischten Schicht}
    xrpp,      {Grundumsatz der Phytoplanktonrespiration}
    xsedra,    {Sedimentationsrate des Phytoplanktons}
    zgzdu,     {Grazing-Rate bei Dunkelheit}
    zgzdu1,   {ausgeklammerter Teil zur Errechnung von zgzdu}
    zgzli,     {Grazing-Rate bei Licht}
    zgzra,     {Grazing-Rate des Zooplanktons}
    zak,       {Verhältnis der assimilierten zur gefressenen Nahrungsmenge des Zooplanktons}
    zassim,    {Assimilationsrate des Zooplanktons}
    zresp,     {Respirationsrate des Zooplanktons}
    zrespt1,   {ausgeklammerter Teil zur Errechnung von zresp}
    zmora,     {Mortalitätsrate des Zooplanktons}
    pexdr,     {Rate der Phosphatremineralisation durch Exkretion des Zooplanktons}
    pzpmort,   {Rate der Phosphatremineralisation aus abgestorbenen oder von Fischen gefresse-
                nem Zooplankton}
    pdkons,    {Phosphatverbrauch durch Phytoplanktonwachstum in der durchmischten Schicht}
    pdremin,   {Phosphatremineralisation in der durchmischten Schicht}
    :real;

procedure licht_und_temperatur; {lokal of model}
const RI      = 0.9; {Reflexionsfaktor}
    RIW      = 0.2; {dto. bei Eis und Schnee}

    FTEMAX    = 4.3; {Wert von FTE, bei dessen Überschreitung eine Abnahme der Lichttrans-
                    mission durch Trübstoffeintrag einsetzt}
    EPSAQUA    = 0.95; {Extinktion durch das reine Wasser und gelöste Substanzen}

var fte: real;      {Faktor für Trübstoffeintrag}

begin
    temp := (tk1 + tk2) + tk1 * sin((wf * tr) + pi * 15/12);
    photper := 12 - 4 * cos(wf * tr);
    li := (Ik * 8.0 * ((280 + 220 * sin(wf * tr + pi * 14/9)) / 24)) * photper;
    if tj[j0,deka] <= 0 then li0 := RIW * li
        else li0 := RI * li;

    {Transmission}
    fte := 5 * (1.5 + 0.5 * cos(wf * tr - pi/6));

    if fte > FTEMAX then lte := 1.2 - ((fte - FTEMAX) / 2)
        else lte := 1.2;
end;

```

```

begin {of procedure model}
  licht_und_temperatur;
  pdimp := pin * qdin;
  xbprjt1 := XBPRMIN + ((XBPRMAX - XBPRMIN) / xgtopt) * temp;
  xrpp := 0.02 * temp;
  zgzdut1 := 2.25 * (exp(-2.0 * abs(ln(temp/zgztot))) + 0.25;
  zrespt1 := (ZRESPOPT - ZRSPTMIN) * sqr((temp/zgztot)) + ZRSPTMIN;
  eps := EPSAQUA - (lte - XEXT * xr) / 2;
  kx := KXP1 * exp((ln(pr)) * KXP2);
  IOJ := ((ln(KI+li0)) - (ln(KI+(li0*exp(-eps*zmixd)))) / (eps*zmixd);
  xbpr := xbprjt1 * ((pr / xr) / (KXP/kx + pr/kx + KXP/xr + pr/xr))
    * IOJ;
  xresp := xrpp + 0.3 * xbpr;
  xgr := xbpr - xresp;
  xg := xgr * xr;
  xsedra := xs / zmixd;
  xsedi := xsedra * xr;
  kz := KZP1 * exp((ln(xr)) * KZP2);
  zgzdu := zgzdut1 * ((xr / zr) / (KGZX/kz + xr/kz + KGZX/zr + xr/zr));
  zgqli := zgzdu * ((24 - photper) / 24) + zgzi * zgdu * (photper / 24);

  if (deka >= sst) and (deka < hvz) then zgza := zgqli
    else zgza := zgzdu;

  zg := zgza * zr;
  dx := xg - xsedi - zg;
  zak := (ZAKMAX - ((ZAKMAX - ZAKMIN) / ZGZMAX) * zgza) * wz;
  zassim := zak * zgza;
  zresp := (((ZRESPOPT - ZRESPMIN) / ZGZMAX) * zgza + ZRESPMIN)
    * (1/ZRESPOPT) * zrespt1;
  zgr := zassim - zresp;
  zg := zgr * zr;
  zmora := (ZMOMIN + ZMOT * temp) * (zr / (KZMO + zr));
  zmort := zmora * zr;
  dz := zg - zmort;

  pdexp := pr * qdin; {Qout = qdin}
  pdkons := (xgr/kznup) * xr * vold;
  pzmort := (0.5 * zmora) / kznup;
  pexkr := (zgza/kznup - zgr/kznup) * 0.5;
  pdremin := (pzmort + pexkr) * zr * vold;
  dp := (pdimp - pdexp - pdkons + pdremin) / vold;
end;

begin
  for j := 1 to 4 do
    begin
      if j = 1 then
        begin
          xr := x;
          zr := z;
          pr := pd;
          tr := t;
        end;
    end;
end;

```

```

    if (j = 2) or (j = 3) then
    begin
        xr := x + 0.5* k[j-1];
        zr := z + 0.5* l[j-1];
        pr := pd + 0.5* m[j-1];
        tr := t + (h/2);
    end;
    if j = 4 then
    begin
        xr := x + k[j-1];
        zr := z + l[j-1];
        pr := pd + m[j-1];
        tr := t + h;
    end;
    model;
    k[j] := dx;
    l[j] := dz;
    m[j] := dp;
end;
x := x + (k[1] + 2 * k[2] + 2 * k[3] + k[4]) * (h/6);
z := z + (l[1] + 2 * l[2] + 2 * l[3] + l[4]) * (h/6);
pd := pd + (m[1] + 2 * m[2] + 2 * m[3] + m[4]) * (h/6);
end;

{ ----- Berechnung der PO4-Remineralisation ----- }
procedure remineralisation;
const optimfakt = 0.65;           {Optimierungsparameter}
var   fdiff,                      {Phosphatfreisetzung aus dem Sediment}
      pfdiff                      {Umrechnung in Relation zur Sedimentflaeche}
      : real;

begin
    fdiff := ((1.21 * (9600 - ph)) - 1700) * 0.002; {Schichtdicke 0.02 m}
    pfdiff := fdiff * (sedarea / (vol - vole_5m)) * 10 * optimfakt;
    ph := ph + pfdiff;
end;

{ ----- Durchmischungspfeile in arrays einlesen ----- }
procedure array_flash_vz;
var i,x,y: integer;
    datei_flash_vz: file of integer;

begin
    {read datei_flash_vz von Diskette/Festplatte}
    assign(datei_flash_vz, 'FLASH_VZ.DAT');
    reset(datei_flash_vz);

```

```

while not eof(datei_flash_vz) do
  begin
    for i := 255 to 390 do
      begin
        read(datei_flash_vz,x);
        flashx1[i] := x;
        read(datei_flash_vz,y);
        flashy1[i] := y;
      end;
    for i := 450 to 555 do
      begin
        read(datei_flash_vz,x);
        flashx2[i] := x;
        read(datei_flash_vz,y);
        flashy2[i] := y;
      end;
    for i := 315 to 495 do
      begin
        read(datei_flash_vz,x);
        flashx3[i] := x;
        read(datei_flash_vz,y);
        flashy3[i] := y;
      end;
    for i := 699 to 804 do
      begin
        read(datei_flash_vz,x);
        flashx4[i] := x;
        read(datei_flash_vz,y);
        flashy4[i] := y;
      end;
    for i := 864 to 999 do
      begin
        read(datei_flash_vz,x);
        flashx5[i] := x;
        read(datei_flash_vz,y);
        flashy5[i] := y;
      end;
    for i := 756 to 936 do
      begin
        read(datei_flash_vz,x);
        flashx6[i] := x;
        read(datei_flash_vz,y);
        flashy6[i] := y;
      end;
    end;
  close(datei_flash_vz);
end;

```

```

{----- PO4 - Pfeile (Remineralisation) -----}
procedure phosphat_pfeile(color:boolean);
var copfeil:           byte;
    pfeilx1,pfeilx2,pfeilyanf,pfeilyend: integer;

procedure pfeil(xpos,yanf,yend:integer; co,p:byte);
begin
    vline(xpos,yanf,yend,co,0);
    pset((xpos-1),(yend+1),co,p);
    pset((xpos+1),(yend+1),co,p);
    pset((xpos-2),(yend+2),co,p);
    pset((xpos+2),(yend+2),co,p);
    pset((xpos-3),(yend+3),co,p);
    pset((xpos+3),(yend+3),co,p);
    PP((xpos-10),(yend-7),co,p);
    OO((xpos-3),(yend-7),co,p);
    z4((xpos+6),(yend-4),co,p);
end;

begin
    pfeilx1 := win[3,xol]+165;
    pfeilx2 := win[3,xol]+235;
    Pfeilyanf := win[3,yur]-4;
    Pfeilyend := win[3,yur]-12;
    if color = true then copfeil := 15
        else copfeil := pointcolor(pfeilx1-10,pfeilyend-8,0);
    pfeil(pfeilx1,pfeilyanf,pfeilyend,copfeil,0);
    pfeil(pfeilx2,pfeilyanf,pfeilyend,copfeil,0);
end;

{----- Grafikausgabe pro Dekade -----}
procedure output_grafik;
var i,j,
    xanf,yanf,hxanf,hyanf,wert,
    y,yn,x1,y1,x2,y2,x11,x12,
    xpoint,xipoint:           integer; { Laufvariable }
    ab,co,co2,co3,co4,co5,co6,co7,
    co8,co10,co13,co14,co15:   byte;   { div. Koordinaten }
    xkorrr,zkorrr:             real;     { " }
    tfe:                       jz;       { " }

    {Farbwahl}
    {Korrekturfaktoren}
    {Tiefenschicht}

procedure eis_und_schnee (xkoo,ykoo: integer; zab,f,p:byte); {lokal}
begin
    EE (xkoo,ykoo,f,p); II (xkoo+1*zab,ykoo,f,p);
    SS (xkoo+2*zab,ykoo,f,p); klein_u(xkoo+4*zab,ykoo,f,p);
    klein_n(xkoo+5*zab,ykoo,f,p); klein_d(xkoo+6*zab,ykoo,f,p);
    SS (xkoo+8*zab,ykoo,f,p); CC (xkoo+9*zab,ykoo,f,p);
    HH (xkoo+10*zab,ykoo,f,p); NN (xkoo+11*zab,ykoo,f,p);
    EE (xkoo+12*zab,ykoo,f,p); EE (xkoo+13*zab,ykoo,f,p);
end;

```

```

procedure O2defizit(tfesub: jz);
var xO2: integer;           {Uebersaettigung im mg/l}

begin
  sed := round(cumsed * 10);

  {Festlegung des Uebergangs des Modellsees in das Stadium des Umkippens}
  case sed of
    42..45: endstadium1 := true;
    46..49: begin
      endstadium1 := false;
      endstadium2 := true;
    end;
  end;

  if sed > 50 then begin
    endstadium1 := false;
    endstadium2 := false;
    endstadium3 := true;
  end;

  {Berechnung der Sauerstoffuebersaettigung bzw. des -defizits}
  case tfesub of
    j0: if ( (deka > 33) and (deka <= 35) and (endstadium3 = true) )
      then odef[tfesub] := -10.0
      else odef[tfesub] := 0.0;

    j5: if ( (deka > 33) and (deka <= 35) )
      then if endstadium3 = true then odef[tfesub] := -10.0
            else begin
                  end
            end
      else
        begin
          xO2 := round(x);
          case xO2 of
            0..4: odef[tfesub] := 0.0;
            5..7: odef[tfesub] := 1.0;
            7..9: odef[tfesub] := 2.0;
            9..11: odef[tfesub] := 3.0;
            11..13: odef[tfesub] := 3.5;
          end;
          if xO2 > 13 then odef[tfesub] := 4.0;
        end;

    j10: if ( (deka > 33) and (deka <= 35) )
      then begin
        if endstadium2 = true then odef[tfesub] := -2.0;
        if endstadium3 = true then odef[tfesub] := -10.0;
      end
      else odef[tfesub] := 0.0;
  end;

```

```

j15: if ( (deka > 33) and (deka <= 35) )
  then begin
    if endstadium1 = true then odef[tfesub] := -2.0;
    if endstadium2 = true then odef[tfesub] := -4.0;
    if endstadium3 = true then odef[tfesub] := -10.0;
  end
else
  begin
    case sed of
      0,1 : odef[tfesub] := 0.0;
      2..6 : odef[tfesub] := -0.5;
      7..10: odef[tfesub] := -1.0;
      11..14: odef[tfesub] := -1.5;
      15..17: odef[tfesub] := -2.0;
      18..20: odef[tfesub] := -2.5;
      21..23: odef[tfesub] := -3.0;
      24,25: odef[tfesub] := -3.5;
      26,27: odef[tfesub] := -4.0;
      28,29: odef[tfesub] := -4.5;
      30: odef[tfesub] := -5.0;
      31: odef[tfesub] := -5.5;
      32: odef[tfesub] := -6.0;
      33: odef[tfesub] := -6.5;
      34: odef[tfesub] := -7.0;
      35: odef[tfesub] := -7.5;
      36..43: odef[tfesub] := (28.0 - sed);
      44..49: odef[tfesub] := (28.0 - (sed + sed - 43));
    end;
    if sed > 49 then odef[tfesub] := (28.0 - (5 * sed - 190));
  end;
end;

j20: if ( (deka > 33) and (deka <= 35) )
  then begin
    if endstadium1 = true then odef[tfesub] := -4.0;
    if endstadium2 = true then odef[tfesub] := -6.0;
    if endstadium3 = true then odef[tfesub] := -10.0;
  end
else
  begin
    case sed of
      0,1 : odef[tfesub] := 0.0;
      2..4 : odef[tfesub] := -1.0;
      5,6 : odef[tfesub] := -1.5;
      7,8 : odef[tfesub] := -2.0;
      9,10: odef[tfesub] := -2.5;
      11,12: odef[tfesub] := -3.0;
      13,14: odef[tfesub] := -3.5;
      15,16: odef[tfesub] := -4.0;
      17,18: odef[tfesub] := -4.5;
      19,20: odef[tfesub] := -5.0;
      21,22: odef[tfesub] := -5.5;
      23,24: odef[tfesub] := -6.0;
      25,26: odef[tfesub] := -6.5;
      27: odef[tfesub] := -7.0;
      28: odef[tfesub] := -7.5;
    end;
  end;
end;

```

```

29: odef[tfesub] := -8.0;
30: odef[tfesub] := -8.5;
31: odef[tfesub] := -9.0;
32: odef[tfesub] := -9.5;
33..39: odef[tfesub] := (23.0 - sed);
40..46: odef[tfesub] := (23.0 - (sed + sed - 39));
47: odef[tfesub] := -33.0;
48: odef[tfesub] := -36.0;
49: odef[tfesub] := -40.0;
end;
if sed > 49 then odef[tfesub] := -50.0;
end;

j25: if ( (deka > 33) and (deka <= 35) )
then begin
    if endstadium1 = true then odef[tfesub] := -4.0;
    if endstadium2 = true then odef[tfesub] := -6.0;
    if endstadium3 = true then odef[tfesub] := -10.0;
end
else
begin
    case sed of
        0,1 : odef[tfesub] := 0.0;
        2 : odef[tfesub] := -2.0;
        3..28: odef[tfesub] := 0.0 - (sed - 1);
        29..34: odef[tfesub] := (0.0 - (sed + sed - 29));
        35: odef[tfesub] := -42.0;
        36: odef[tfesub] := -45.0;
        37: odef[tfesub] := -48.0;
        38: odef[tfesub] := -52.0;
        39: odef[tfesub] := -56.0;
        40: odef[tfesub] := -60.0;
        41: odef[tfesub] := -65.0;
        42: odef[tfesub] := -70.0;
    end;
    if sed > 43 then odef[tfesub] := -75.0;
end;
end; {of case tfe sub }
end; {of procedure O2defizit}

```

```

procedure draw_arrays (xsub1,ysub1,xsub2,ysub2,xsub1,xsub2,np: integer);
{lokal}
{
    xsub1,ysub1: Koordinaten des Startpunktes (osat)
    xsub1,ysub1:      "      "      "      (odef)
    xsub2,ysub2: Koordinaten des Endpunktes (osat)
    xsub2,ysub2:      "      "      "      (odef)
    np: Zahl der Punkte in y-Richtung
}
var i,diffx,diffxd,xpoint,xdpoint,col,co2: integer;

begin
    col := 11; {Kurve O2-sat}
    diffx := xsub1 - xsub2;
    for i := ysub1 to ysub2 do
        begin
            xpoint := round(xsub1 - (i-(ysub1-1)) * (diffx / np)) + 1;
            points_osat[i] := xpoint;
            pset(xpoint,i,col,0);
        end;

    co2 := 7; {Kurve O2-def}
    diffxd := xsub1 - xsub2;
    for i:= ysub1 to ysub2 do
        begin
            xdpoint := round(xsub1 - (i-(ysub1-1)) * (diffxd / np));
            if xdpoint < (win[4,xol]+1) then xdpoint := win[4,xol]+1;
            points_oakt[i] := xdpoint;
            pset(xdpoint,i,co2,0);
        end;
    end;

procedure o2_schraffierung(col,co2,co3: byte); {lokal}
{col: Farbe wenn O2-Defizit > 0.0 mg O2/l
  co2: Farbe Uebersaettigung
  co3: Farbe wenn O2-Defizit = 0.0 mg O2/l}

begin
    for i := (win[4,yol]+5) to (win[4,yur]-1) do
        begin
            if (points_oakt[i] + 1) < points_osat[i]
            then
                if points_oakt[i] = (win[4,xol]+1)
                then
                    hline(points_osat[i]-1,points_oakt[i],i,co3,0)
                else
                    hline(points_osat[i]-1,points_oakt[i]+1,i,col,0);
            if points_oakt[i] > points_osat[i] + 2 then
                hline(points_osat[i]+1,points_oakt[i]-1,i,co2,0);
        end;
    end;
end;

```

```
procedure select_monat; {lokal}
```

```
begin
```

```
  case deka of
```

```
    1: monat := '  Januar';
```

```
    4: monat := ' Februar';
```

```
    7: monat := '   März';
```

```
   10: monat := ' April';
```

```
   13: monat := '   Mai';
```

```
   16: monat := '   Juni';
```

```
   19: monat := '   Juli';
```

```
   22: monat := ' August';
```

```
   25: monat := 'September';
```

```
   28: monat := ' Oktober';
```

```
   31: monat := ' November';
```

```
   34: monat := ' Dezember';
```

```
end;
```

```
end;
```

```
begin {of procedure output_grafik}
```

```
  co2 := 2; {Schraff. Uebersaettigung}
```

```
  co3 := 3; {Jahr, Monat}
```

```
  co4 := 4; {Schraff. O2-defizit anaerob}
```

```
  co5 := 5; {Schraff. O2-defizit aerob}
```

```
  co6 := 6; {Zooplankton}
```

```
  co7 := 7; {Kurve O2-def}
```

```
  co8 := 11; {Kurve O2-sat}
```

```
  co10 := 10; {Algen}
```

```
  co13 := 13; {Licht 10J5m}
```

```
  co14 := 14; {Licht über Wasseroberfl.}
```

```
  co15 := 15; {Eis, Text Eis, Wassertemp.}
```

```
{Monat}
```

```
select_monat;
```

```
writepage(0); textattr(co3,0); e_gotoxy(71,2,0); write(monat);
```

```

{draw and write Eis und Schnee}
if (tj[j0,deka] = 0) and (drawflag = false)
then
  begin {draw and write}
    xanf := win[3,xol] + 240;
    yanf := win[3,yol] - 5;
    ab := 6;
    hline((win[3,xol]+3), (win[3,xur]-2), (win[3,yol]+2), col5, 0);
    hline((win[3,xol]+4), (win[3,xur]-3), (win[3,yol]+3), col5, 0);
    eis_und_schnee(xanf, yanf, ab, col5, 0);
    drawflag := true;
  end
else
  begin {loesche}
    if (tj[j0,deka] <> 0) and (drawflag = true) then
      begin
        xanf := win[3,xol] + 240;
        yanf := win[3,yol] - 5;
        ab := 6;
        hline((win[3,xol]+3), (win[3,xur]-2), (win[3,yol]+2), 0, 0);
        hline((win[3,xol]+4), (win[3,xur]-3), (win[3,yol]+3), 0, 0);
        eis_und_schnee(xanf, yanf, ab, 0, 0);
        drawflag := false;
      end;
    end;
  end;

{Temperaturschichtung}
if klimasign = 1 then schraffierung1
  else if klimasign = 2 then schraffierung2
    else schraffierung3;

{tiefenverteilte Wassertemperatur}
writepage(0); textattr(col5, 0);
e_gotoxy(39, 10, 0); write(tj[j0 ,deka]:2, '°C');
e_gotoxy(39, 11, 0); write(tj[j5 ,deka]:2, '°C');
e_gotoxy(39, 13, 0); write(tj[j10,deka]:2, '°C');
e_gotoxy(39, 16, 0); write(tj[j15,deka]:2, '°C');
e_gotoxy(39, 18, 0); write(tj[j20,deka]:2, '°C');
e_gotoxy(39, 21, 0); write(tj[j25,deka]:2, '°C');

```

```

{Werteausgabe}
writepage(0); textattr(col5,0);
e_gotoxy( 2, 8,0); write(pin:5:1);{P-Imp.}
e_gotoxy(67, 8,0); write( pd:5:1);{P-Exp.}
e_gotoxy(11,42,0); write( li:4:0);{Licht 1}
e_gotoxy(11,43,0); write(IQJ5m:4:0);{Licht 2}
e_gotoxy(29,43,0); write( pd:5:1);{P}
e_gotoxy(49,43,0); if x < 0.01 then begin
                                {X}
                                xkorr := 0.01;
                                write(xkorr:5:2);
                                end
                                else write(x:5:2);

e_gotoxy(69,43,0); if z < 0.01 then begin
                                {Z}
                                zkorr := 0.01;
                                write(zkorr:5:2);
                                end
                                else write(z:5:2);

{O2 - Saettigungskurve}
yanf := win[4,yol] + 5; {erster moeglicher y-Punkt}
xanf := win[4,xol] + 1; {Beginn der Punkte in x-Richtung}
yn := trunc((win[4,yur] - yanf) / 5); {Zahl der Punkte in y-Richtung}

{loesche Sauerstoffkurven der vorigen Dekade}
if deka > 1 then
begin
for i := yanf to (win[4,yur]- 1) do
begin
xpoint := points_osat[i];
pset(xpoint,i,0,0);
xdpoint := points_oakt[i];
pset(xdpoint,i,0,0);
end;
{loesche O2-Schraffierung der vorigen Dekade}
o2_schraffierung(0,0,0);
end;

```

```

(draw Sauerstoffkurven)
tfe := j0;
i := 0;
repeat
  {x-Koordinaten der Sättigungskurve}
  x1 := xanf + round(osat[tfe,deka] * 8);
  x2 := xanf + round(osat[succ(tfe),deka] * 8);

  {x-Koordinaten der aktuellen O2-Kurve}
  case deka of 8,9,34..36:
    begin
      if ( (endstadium1 = false) and (endstadium2 = false)
          and (endstadium3 = false) ) then odef[tfe] := 0.0
          else O2defizit(tfe);
        end
      else O2defizit(tfe);
    end;

  xd1 := xanf + round((osat[tfe,deka] + odef[tfe]) * 8);
  if xd1 > win[4,xur] then xd1 := win[4,xur];
  if xd1 < xanf then xd1 := xanf;

  case deka of 8,9,34..36:
    begin
      if ( (endstadium1 = false) and (endstadium2 = false)
          and (endstadium3 = false) ) then odef[succ(tfe)] := 0.0
          else O2defizit(succ(tfe));
        end
      else O2defizit(succ(tfe));
    end;

  xd2 := xanf + round((osat[succ(tfe),deka] + odef[succ(tfe)]) * 8);

  y1 := yanf + i * yn; {Anfangspunkt im Punktefeld}
  y2 := yanf + (yn-1) + i * yn; {Endpunkt im Punktefeld}
  draw_arrays(x1,y1,x2,y2,xd1,xd2,yn);

  tfe := succ(tfe);
  i := i + 1;
until tfe > j20;

```

```

{schraffierte Defizit bzw. Uebersaettigung}
o2_schraffierung(co5,co2,co4);

{Ausgabe der Histogramme}
for j := 5 to 8 do
begin
  hxfanf := (win[j,xol]+XALI+1) + versetz;
  hyanf := win[j,yur]-YAUN-1;
  case j of
    5: begin {li}
        wert := hyanf - round(li * 0.03);
        co := co14;
      end;
    6: begin {P}
        wert := hyanf - round(pd * 1.5);
        co := co15;
      end;
    7: begin {X}
        wert := (hyanf - round(x * 5)) - wertoverxmax; {xmax=110 points};
        wertoverxmax := 0;
        wertoverxmax := 199 - wert; {hyanf-110=199}
        if wertoverxmax > 0 then wert := 199
        else wertoverxmax := 0;
        co := co10;
      end;
    8: begin {Z}
        wert := (hyanf - round(z * 8)) - wertoverzmax; {zmax=104 points}
        wertoverzmax := 0;
        wertoverzmax := 205 - wert; {hyanf-104=205}
        if wertoverzmax > 0 then wert := 205
        else wertoverzmax := 0;
        co := 6;
      end;
  end;
  for i := 0 to 3 do
    vline(hxfanf+i,hyanf,wert,co,0);
  end;

  {Licht in 3 m Tiefe}
  hxfanf := (win[5,xol]+XALI+1+1) + versetz;
  hyanf := win[5,yur]-YAUN-1;
  wert := hyanf - round(IOJ5m * 0.03);
  co := 13;
  for i := 0 to 1 do
    vline(hxfanf+i,hyanf,wert,co,0);
  end;

```

```

{Hinweise}
{interne DÜngung}
if odef[j25] >= -15
  then if flag_intd = true
    then begin
      interne_duengung(false);
      flag_intd := false;
    end;
  if (odef[j25] < -15) and (odef[j25] >= - 50)
    then if (flag_intd = false) { and (flag_blau = false) }
      and (flag_h2s = false)
        then begin
          interne_duengung(true);
          flag_intd := true;
        end;

{H2S}
if odef[j25] >= -40
  then if flag_h2s = true
    then begin
      hinweis_H2S(false);
      flag_h2s := false;
    end;
  if odef[j25] < -40
    then if flag_h2s = false
      then begin
        hinweis_H2S(true);
        flag_h2s := true;
      end;
end;

```

```

===== Anweisungsteil =====
begin
  ClrScr;
  text_80x25;
  text_bypass;
  colorpalette(0,0);
  writepage(0);
  selectpage(0);
  logo; {OVR-procedure}
  delay(3000);

  {Festlegen der Pseudo-Window-Koordinaten}
  win[1,xol]:=120; win[1,yol]:= 11; win[1,xur]:=519; win[1,yur]:= 56; {Hinw.}
  win[2,xol]:= 0; win[2,yol]:= 67; win[2,xur]:=119; win[2,yur]:=167; {Sicht.}
  win[3,xol]:=120; win[3,yol]:= 67; win[3,xur]:=519; win[3,yur]:=167; {Seeb.}
  win[4,xol]:=520; win[4,yol]:= 67; win[4,xur]:=639; win[4,yur]:=167; {Oxyg.}
  win[5,xol]:= 0; win[5,yol]:=193; win[5,xur]:=159; win[5,yur]:=349; {Licht}
  win[6,xol]:=160; win[6,yol]:=183; win[6,xur]:=319; win[6,yur]:=349; {Phos.}
  win[7,xol]:=320; win[7,yol]:=183; win[7,xur]:=479; win[7,yur]:=349; {Algen}
  win[8,xol]:=480; win[8,yol]:=193; win[8,xur]:=639; win[8,yur]:=349; {Zoopl.}

  {fill seebeckenarray mit sinuskurve}
  ClrScr;
  e_gotoxy(27,10,0); textattr(14,0); writeln('Bitte einen Moment Geduld. ');
  e_gotoxy(27,12,0); writeln('Es werden Daten eingelesen. ');
  e_gotoxy(1,43,0); {Cursor hidden}
  read_datei_waterline; {fill wly_array mit Koordinaten von Massenspeicher}
  array_flash_vz; {fill flasharray mit Punkte-Koordinaten von Massenspeicher}

  res_640x350;

  morphometrie; {OVR-procedure mit lokaler procedure vole_j}

  {Anfangsbedingungen}
  ew := 300;
  pzsatz := 100;
  lwfl := 200;
  entsorg := 'A';

  xgtopt := 20.0; {°C}
  zgztot := 20.0; {°C}
  kmup := 0.8; {cm**3/mg P}
  kmup := 0.8; {cm**3/mg P}
  uxz := 1.0;
  zgzi := 0.33;
  xs := 0.1; {m/d}

  jahr := 1;

  flag_abbruch := false;
  endstadium1 := false;
  endstadium2 := false;
  endstadium3 := false;

```

```

menu_laufotyp; {OVR-procedure}
eingabe_trophie; {OVR-procedure}

repeat
{----- pro Jahr -----}

t      := TANF;
h      := 2;
deka   := 1;
time   := 1;
wertoverxmax := 0;
wertoverzmax := 0;
drawflag := false;
flag_intd := false;
flag_h2s := false;
pfeilflag := false;
flag_last := false;
versetz := 0;

odef[ j0] := 0.0; odef[ j5] := 0.0; odef[j10] := 0.0;
odef[j20] := 0.0; odef[j25] := 0.0;

eingabe_klima; {OVR-procedure}
phosphatbilanz; {OVR-procedure}
draw_grafik; {OVR-procedure}
e_gotoxy(1,2,0); writepage(0); textattr(3,0); write(jahr:3,'.Jahr');

repeat
{----- pro Zeitschritt -----}
pruefstop;
durchmischungstiefe;
runge_kutta;
time := trunc(t/10);
if (t >= time * 10) and (time = deka) then
begin
{----- pro Dekade -----}
{Anf.bed. pro Jahr speichern}
if deka = 1 then begin
xanf := x;
zanf := z;
pdanf := pd;
cumsedanf := cumsed;
end;

{Kumulation der Sedimentationsraten von X und Z}
cumsed := cumsed + xsedi + (zmort / 3);
if deka = 9 then cumsed := cumsed * 0.8; {Red. bei FVZ}
if ( (deka = 34) and (endstadium1 = false) and (endstadium2 = false)
and (endstadium3 = false) )
then cumsed := cumsed * 0.6; {Red. bei HVZ}

```

```

if (odef[j25] < -14)
  then begin
    remineralisation;
    phosphat_pfeile(true);
    pfeilflag := true;
  end;
if (odef[j25] >= -14) and (pfeilflag = true)
  then begin
    phosphat_pfeile(false); {zuruecksetzen in der
                             richtigen Farbe}
    pfeilflag := false;
  end;

if deka = 9 then ph := pd; {Maerz-Wert von pd nach ph}
if deka = hvz then
  pd := (pd * vole_5m + ph * (vol - vole_5m)) / vol; {Rückmischung}

{IOJ in 5m Tiefe}
IOJ5m := (((ln(KI+li0))-ln(KI+(li0*exp(-eps*5))))/(eps*5))
          * li0;

{Pruefen, ob Abbruch}
if ( (deka > 33) and (deka <=36) and (endstadium3 = true) )
  then begin
    hinweis_abbruch(true);
    flag_last := true;
    deka := 36;
  end;

  deka := deka + 1;
  versetz := versetz + 4;
end;

if (abs(xgr) >= 0.003) then h := 2
  else h := 4;

t := t + h;

until deka > 36;

{loesche alle Hinweise}
if flag_intd = true then interne_duengung(false);
if flag_h2s = true then hinweis_H2S(false);
e_press_key(25,4,0);
jahr := jahr + 1;

menu_weiter;

until flag_abbruch = true;      { Abbruchbedingung }
end.

```

```

===== INTRO.OVL =====
Diese Datei enthaelt Vereinbarungen ueber diverse Prozeduren
a) fuer die Erstellung der Grafik,
b) fuer die Berechnung verschiedener Volumina,
c) fuer die verschiedenen Auswahlmenues
=====

```

```

{-----}
OVERLAY procedure logo;
begin
textattr(11,0);
e_gotoxy( 3, 8,0);write('');
e_gotoxy( 3, 9,0);write('');
e_gotoxy( 3,10,0);write('');
e_gotoxy( 3,11,0);write('');
e_gotoxy( 3,12,0);write('');
e_gotoxy( 3,13,0);write('');
e_gotoxy( 3,14,0);write('');
e_gotoxy( 3,15,0);write('');
e_gotoxy( 3,16,0);write('');
textattr(11,0);
e_gotoxy( 3,17,0);write('Copyright: Peter G. Hiering');
e_gotoxy(1,26,0);
end;

{----- Daten fuer Seebecken aus Datei WLY.DAT einlesen -----}
OVERLAY procedure read_datei_waterline;
var  xpoint,half,full:   integer;
      datei_wly:         file of integer;

begin
  assign(datei_wly,'WLY.DAT');
  reset(datei_wly);
  xpoint := 0;
  while not eof(datei_wly) do
    begin
      xpoint := xpoint + 1;
      read(datei_wly,wly[xpoint]);
      baseline1[xpoint] := wly[xpoint] + 1;
      baseline2[xpoint] := wly[xpoint] + 2;
      baseline3[xpoint] := wly[xpoint] + 3;
    end;
  close(datei_wly);

```

```

half := trunc((win[3,xur] - win[3,xol] + 1) / 2);
full := (win[3,xur] - win[3,xol] + 1);
for xpoint := 1 to half do
  if (xpoint = 1) or (wly[xpoint] <> wly[xpoint-1])
  then hxl[wly[xpoint]] := xpoint;
for xpoint := full downto (half + 1) do
  if (xpoint = full) or (wly[xpoint] <> wly[xpoint+1])
  then hxr[wly[xpoint]] := xpoint;
end;

{----- Auswahl des Anfangszustand des Sees -----}
OVERLAY procedure eingabe_trophie;
var zeichen: char;

begin
  text_80x25;
  text_bypass;
  writepage(0);
  textattr(14,0);
  e_gotoxy(15, 3,0);write('ANFANGSZUSTAND DES SEES zu Beginn der Simulation:');
  e_gotoxy(15, 4,0);write(' (zur Auswahl)');
  e_gotoxy(15, 7,0);write('nährstoffarm ..... 1');
  e_gotoxy(15, 8,0);write(' (oligotroph)');
  e_gotoxy(15,10,0);write('leicht nährstoffreich ..... 2');
  e_gotoxy(15,11,0);write(' (mesotroph)');
  e_gotoxy(15,13,0);write('nährstoffreich ..... 3');
  e_gotoxy(15,14,0);write(' (eutroph)');
  e_gotoxy(30,18,0);write('Ihre Auswahl:');
  zeichen := ' ';
  flush_keyboard;
  e_gotoxy(44,18,0);
  blink_cursor(44,18,10);
  read(Kbd,zeichen);
  textattr(10,0); write(zeichen);
  delay(350);
  flush_keyboard;
  if zeichen in ['1'..'3']
  then
    case zeichen of
      '1': begin
        x := 0.01; z := 0.1; pd := 1.0; cumsed := 0.2;
      end;
      '2': begin
        x := 0.05; z := 0.2; pd := 15.0; cumsed := 0.5;
      end;
      '3': begin
        x := 0.1; z := 0.3; pd := 40.0; cumsed := 1.2;
      end;
    end
  else begin
    write(Chr(7));
    eingabe_trophie;
  end;
end;

```

```

{----- Auswahl der klimatischen Bedingungen -----}
OVERLAY procedure eingabe_klima;
var zeichen: char;

begin
  text_80x25;
  text_bypass;
  writepage(0);
  textattr(14,0);
  e_gotoxy(20, 3,0); write('KLIMATISCHE BEDINGUNGEN (zur Auswahl)');
  e_gotoxy(20, 7,0); write('warmer Sommer,');
  e_gotoxy(20, 8,0); write('warmer Winter,');
  e_gotoxy(20, 9,0); write('wenig Bewölkung ..... 1');
  e_gotoxy(20,11,0); write('normaler Sommer,');
  e_gotoxy(20,12,0); write('normaler Winter ..... 2');
  e_gotoxy(20,14,0); write('kühler Sommer,');
  e_gotoxy(20,15,0); write('kalter Winter,');
  e_gotoxy(20,16,0); write('häufige Bewölkung ..... 3');
  e_gotoxy(33,21,0); write('Ihre Auswahl:');
  zeichen := ' ';
  flush_keyboard;
  e_gotoxy(48,21,0);
  blink_cursor(48,21,10);
  read(Kbd,zeichen);
  e_gotoxy(48,21,0); textattr(10,0); write(zeichen);
  delay(350);
  flush_keyboard;
  if zeichen in ['1'..'3']
  then
    case zeichen of
      '1': begin
        tk1 := 10.5; tk2 := 1.5; Ik := 1.3;
        sst := 13; hvz := 34;
        klimasign := 1;
        temperaturverteilung_1;
      end;
      '2': begin
        tk1 := 9.0; tk2 := 1.5; Ik := 1.0;
        sst := 14; hvz := 34;
        klimasign := 2;
        temperaturverteilung_2;
      end;
      '3': begin
        tk1 := 8.0; tk2 := 0.5; Ik := 0.7;
        sst := 17; hvz := 34;
        klimasign := 3;
        temperaturverteilung_3;
      end;
    end;
  end
end

```

```

else begin
    write(Chr(7));
    eingabe_klima;
end;
sauerstoffsattigung;
end;

{----- Berechnung der Phosphatbilanz -----}
OVERLAY procedure phosphatbilanz;
const col = 15; {Haupttext}
      co2 = 11; {Hervorhebung 1}
      co3 = 10; {Hervorhebung 2}
      co4 = 14; {Programmanweisung}

type asctype = string[4];

var neustring: string[80];

OVERLAY procedure text_seedorf; {lokal zu phosphatbilanz}
begin
    lines 43;
    text_bypass;
    writepage(0); selectpage(0);
    e_gotoxy(55, 1,0); textattr(col,0); write('*');
    e_gotoxy(26, 2,0); write('P H O S P H A T - I M P O R T');
    e_gotoxy(26, 3,0); write('=====');
    e_gotoxy(28, 4,0); write('aus der Gemeinde SEEDORF');
    e_gotoxy( 1, 7,0); textattr(co2,0); write('EINWOHNERZAHL');
    e_gotoxy(15, 7,0); textattr(col,0); write('der Gemeinde SEEDORF ... :');
    e_gotoxy( 1, 8,0); write('Prozentsatz der Einwohner, die');
    e_gotoxy( 1, 9,0); textattr(co2,0); write('phosphathaltige Waschmittel');
    e_gotoxy(29, 9,0); textattr(col,0); write('benutzen . :');
    e_gotoxy(47, 9,0); textattr(co2,0); write('%');
    e_gotoxy( 1,10,0); textattr(col,0); write('direkt an den See grenzende,');
    e_gotoxy( 1,11,0); textattr(co2,0); write('gedüngte WIESEN und ACKERFLÄCHEN ..... :');
    e_gotoxy(47,11,0); write('ha');
    e_gotoxy( 1,13,0); write('Behandlung des ABWASSERS');
    e_gotoxy(26,13,0); textattr(col,0); write('nach Situation:');
    e_gotoxy( 3,14,0); write('zur Auswahl:');
    e_gotoxy( 4,15,0); write('Die Abwässer werden in den See geleitet:');
    e_gotoxy( 4,16,0); textattr(co3,0); write('A: direkt');
    e_gotoxy( 7,17,0); textattr(col,0); write('über eine');
    e_gotoxy(17,17,0); textattr(co3,0); write('KLÄRANLAGE');
    e_gotoxy(28,17,0); textattr(col,0); write('mit der(n) Reinigungsstufe(n)');
    e_gotoxy( 4,18,0); textattr(co3,0); write('B: mechanisch');
    e_gotoxy( 4,19,0); write('C: mechanisch und biologisch');
    e_gotoxy( 4,20,0); write('D: mechanisch, biologisch und chemisch');
    e_gotoxy(45,22,0); textattr(col,0); write('***');
    e_gotoxy(34,23,0); write('B I L A N Z');
    e_gotoxy(34,24,0); write('=====');
    e_gotoxy( 5,25,0); write('atmosphärische Fracht (aus Niederschlägen) .....');
:');
    e_gotoxy(63,25,0); write('g P/Tag');
    e_gotoxy( 5,26,0); write('Abwässer aus den Haushalten .....');
:');

```

```

e_gotoxy(63,26,0);
e_gotoxy( 5,27,0);
:');
e_gotoxy(63,27,0);
e_gotoxy(56,28,0);
e_gotoxy(48,29,0);
e_gotoxy(63,29,0);
e_gotoxy( 1,31,0); write('Bei einer Zuflußmenge von ',qdin:6:0,' m3/Tag ergibt sich');
e_gotoxy( 1,32,0); write('eine Ortho-Phosphatkonzentration von ..... :');
e_gotoxy(63,32,0);
e_gotoxy( 1,43,0);
Jahresdurchschnittswerte');
{Werteausgabe Teil 1}
textattr(co2,0);
e_gotoxy(42, 7,0); write(ew:4);
e_gotoxy(43, 9,0); write(pzsatz:3);
e_gotoxy(42,11,0); write(lwfl:4);
e_gotoxy(43,13,0); write(entsorg);
end;

OVERLAY procedure berechnung_und_werteausgabe; {lokal zu phosphatbilanz}
const NATFR = 12.0;

var   klaer:                byte;
      ewmit, ewohne:        integer;
      haushalt, auswasch, sumphosp: real;
begin
  {Berechnung}
  case entsorg of
    'A'..'C', 'a'..'c': klaer := 100; {%
    'D', 'd'           : klaer := 10;
  end;
  ewmit   := ew * pzsatz div 100;
  ewohne  := ew - ewmit;
  haushalt := (ewmit * 2.3 + ewohne * 1.9) * klaer / 100;
  auswasch := lwfl * 3.8;
  sumphosp := NATFR + haushalt + auswasch; {g P/d}
  pin     := sumphosp * 1000 / qdin; {mg P/m3 * d}

  {Werteausgabe Teil 2}
  textattr(col,0);
  e_gotoxy(56,25,0); write(NATFR:6:1);
  e_gotoxy(56,26,0); write(haushalt:6:1);
  e_gotoxy(56,27,0); write(auswasch:6:1);
  e_gotoxy(56,29,0); write(sumphosp:6:1);
  textattr(co2,0);
  e_gotoxy(56,32,0); write(pin:6:1);
end;

```

```
procedure eingabefeld(xpos,ypos,co: byte; maxwert: asctype;
                     blink: boolean); {lokal zu phosphatimport}
```

```
begin
  e_gotoxy(xpos,ypos,0); textattr(co,0);
  write('neuer Wert (max. ',maxwert:4,'): ',neustring);
  e_gotoxy(xpos+24,ypos,0);
  if blink = true then blink_cursor(xpos+24,ypos,14);
end;
```

```
OVERLAY procedure aendern; {lokal zu phosphatbilanz}
var i: byte;
```

```
  pruefchar: char;
  ewneu,pzsatzneu,lwflneu,
  fehler: integer;
```

```
OVERLAY procedure einwohner_neu; {lokal zu aendern}
```

```
begin
  neustring := ' ';
  eingabefeld(50,7,10,'1500',true);
  textattr(10,0);
  readln(neustring);
  fehler := 0;
  if (length(neustring) <= 4) and (length(neustring) > 0)
  then begin
    for i := 1 to length(neustring) do
    begin
      pruefchar := copy(neustring,i,1);
      if not (pruefchar in ['0'..'9']) then fehler := 1;
    end;
    if fehler = 0
    then begin
      val(neustring,ewneu,fehler);
      if fehler = 0
      then if ewneu <= 1500 then begin
        ew := ewneu;
        exit;
      end;
    end;
  end;
  write(Chr(7));
  einwohner_neu;
end;
```

OVERLAY procedure pzsatz_neu; {lokal zu aendern}

begin

 neuststring := ' ';

 eingabefeld(50,9,10,'100',true);

 textattr(10,0);

 readln(neuststring);

 fehler := 0;

 if (length(neuststring) <= 3) and (length(neuststring) > 0)

 then begin

 for i := 1 to length(neuststring) do

 begin

 pruefchar := copy(neuststring,i,1);

 if not (pruefchar in ['0'..'9']) then fehler := 1;

 end;

 if fehler = 0

 then begin

 val(neuststring,pzsatzneu,fehler);

 if fehler = 0

 then if pzsatzneu <= 100 then begin

 pzsatz := pzsatzneu;

 exit;

 end;

 end;

 end;

 write(chr(7));

 pzsatz_neu;

end;

OVERLAY procedure landwflaeche_neu; {lokal zu aendern}

begin

 neuststring := ' ';

 eingabefeld(50,11,10,'500',true);

 textattr(10,0);

 readln(neuststring);

 fehler := 0;

 if (length(neuststring) <= 3) and (length(neuststring) > 0)

 then begin

 for i := 1 to length(neuststring) do

 begin

 pruefchar := copy(neuststring,i,1);

 if not (pruefchar in ['0'..'9']) then fehler := 1;

 end;

 if fehler = 0

 then begin

 val(neuststring,lwflneu,fehler);

 if fehler = 0

 then if lwflneu <= 500 then begin

 lwfl := lwflneu;

 exit;

 end;

 end;

 end;

```

write(Chr(7));
landwflaeche_neu;
end;

```

```

OVERLAY procedure entsorgung_neu; {lokal zu aendern}
begin
  neustring := '          ';
  eingabefeld(50,13,10,'A..D',true);
  textattr(10,0);
  readln(neustring);
  neustring := UpCase(neustring);
  if (length(neustring) = 1)
    then begin
      pruefchar := copy(neustring,1,1);
      if pruefchar in ['A'..'D'] then begin
        entsorg := neustring;
        exit;
      end;
    end;
  write(Chr(7));
  entsorgung_neu;
end;

```

```

begin {of aendern}
  einwohner_neu;
  eingabefeld(50,7, 0, '9999',false); {blink ausschalten sonst wartet}
                                     {Programm auf keypressed}
  e_gotoxy(42,7,0); textattr(11,0); write(ew:4);

  pzsatz_neu;
  eingabefeld(50,9, 0, '100',false); {blink ausschalten sonst wartet}
                                     {Programm auf keypressed}
  e_gotoxy(43,9,0); textattr(11,0); write(pzsatz:3);

  landwflaeche_neu;
  eingabefeld(50,11,0, '999',false); {blink ausschalten sonst wartet}
                                     {Programm auf keypressed}
  e_gotoxy(42,11,0); textattr(11,0); write(lwfl:4);

  entsorgung_neu;
  eingabefeld(50,13, 0, 'A..D',false); {blink ausschalten sonst wartet}
                                     {Programm auf keypressed}
  e_gotoxy(43,13,0); textattr(11,0); write(entsorg);
end;

```

```

OVERLAY procedure berechnungsgrundlagen; {lokal zu phosphatimport}
begin
  text 80x25;
  text_bypass;
  writepage(0);
  textattr(15,0);
  e_gotoxy(29,1,0); write('BERECHNUNGSGRUNDLAGEN');
  e_gotoxy(29,2,0); write('=====');
  e_gotoxy(1, 5,0); write('Phosphat-Import aus Niederschlägen ..... :');
  e_gotoxy(65, 5,0); write('(Festwert)');
  e_gotoxy(1, 9,0); write('Phosphat-Import durch Abwässer aus Haushalten :');
  e_gotoxy(1,11,0); write('■ bei Verwendung phosphathaltiger Waschmittel :');
  e_gotoxy(1,13,0); write('■ ohne Verwendung phosphathaltiger Waschmittel:');
  e_gotoxy(1,17,0); write('Phosphat-Import durch Auswaschung');
  e_gotoxy(1,18,0); write('landwirtschaftlicher Flächen :');
  e_gotoxy(1,21,0); write('Nur Kläranlagen mit chemischer Reinigungsstufe');
  e_gotoxy(48,21,0); write('vermindern die Phosphate');
  e_gotoxy(1,22,0); write('(um ca. 90 %).');
  textattr(11,0);
  e_gotoxy(50, 5,0); write('12 g P / Tag');
  e_gotoxy(50,11,0); write('2,3 g P / Tag pro Einwohner');
  e_gotoxy(50,13,0); write('1,7 g P / Tag pro Einwohner');
  e_gotoxy(50,18,0); write('3,8 g P / Tag pro Hektar');
  textattr(14,0);
  e_gotoxy(29,24,0); write('zurück zur Simulation:');
  e_gotoxy(29,25,0); write('beliebige Taste drücken !');
  repeat until keypressed;
end;

```

```

procedure bypass_ovl_menue; forward;
OVERLAY procedure menue_und_verzweigung; {lokal zu phosphatimport}

```

```

var zeichen:      char;

```

```

begin
  e_gotoxy(21,35,0); textattr(co4,0); write('');
  e_gotoxy(21,36,0); write('Simulation fortsetzen ..... 1');
  e_gotoxy(21,37,0); write('Daten ändern ..... 2');
  e_gotoxy(21,38,0); write('Berechnungsgrundlagen einsehen .. 3');
  e_gotoxy(21,39,0); write('');
  e_gotoxy(21,40,0); write('Ihre Eingabe:');
  e_gotoxy(21,41,0); write('');

  zeichen := ' ';
  flush_keyboard;
  e_gotoxy(48,40,0);
  blink_cursor(48,40,10);
  read(kbd,zeichen);
  textattr(10,0); write(zeichen);
  delay(350);
  flush_keyboard;

```

```

if zeichen in ['1'..'3']
then
  case zeichen of
    '1': exit;
    '2': begin
      aendern;
      berechnung_und_wertausgabe;
      bypass_ovl_menue;
    end;
    '3': begin
      berechnungsgrundlagen;
      text_seedorf;
      berechnung_und_wertausgabe;
      bypass_ovl_menue;
    end;
  end
else begin
  write(Chr(7));
  bypass_ovl_menue;
end;

```

```
end;
```

```

procedure bypass_ovl_menue;
begin
  menue_und_verzweigung;
end;

```

```

begin {procedure phosphatbilanz}
  text_seedorf;
  berechnung_und_wertausgabe;
  menue_und_verzweigung;
end;

```

```

{-----Berechnung des Volumens der durchmischten Schicht -----}
OVERLAY procedure morphometrie;
var fp: real; {Transformationsfaktor}

```

```

OVERLAY function vole_j (jt:byte; fpf:real):real;
{ Berechnung Volumen Epilimnion für Tiefe = jt; lokal
  fpf: Transformationsfaktor}

```

```

begin
  vole_j := (6*(jt/ZMAX) - 3*(1-fpf)*sqr(jt/ZMAX)
    - 2*fpf*(sqr(jt/ZMAX)*(jt/ZMAX)))
    / (3 + fpf);
end;

```

```

begin
  vol := (1.456 * (L/2) * (L*2/6) * ZMAX);
  fp := 6 * 0.464 - 3;
  vole_5m := vol * vole_j(5,fp);
  vole_7m := vol * vole_j(7,fp);
  vole_12m := vol * vole_j(12,fp);
  vole_16m := vol * vole_j(16,fp);
  vole_20m := vol * vole_j(20,fp);
  sedarea := (L/2) * (L*2/6) * 3.14 * 10/100;
  qdin := vol / 720;
  wf := pi / 180; {Periode}
end;

{----- Erstellung der Bildschirmgrafik -----}
OVERLAY procedure draw_grafik;

var co,ab,
    co3,co6,co10,co11,co13,co14,co15:   byte;
    i,j,xanf,yanf:                       integer;

OVERLAY procedure text_kJ (xkoo,ykoo:integer; zab,f,p:byte; kilo: boolean);
{lokal zu draw_grafik}
var xleft: integer;

begin
  xleft := zab; {Verschiebung von JJ nach links, falls k fehlt}
  case kilo of
    true: begin
      klein_k(xkoo,ykoo,f,p);
      xleft := 0; {keine Verschiebung}
    end;
  end;
  JJ      ((xkoo+1*zab) -xleft,ykoo, f,p);
  sstr    ((xkoo+2*zab) -xleft,ykoo, f,p);
  klein_c((xkoo+3*zab) -xleft,ykoo, f,p);
  klein_m((xkoo+4*zab) -xleft,ykoo, f,p);
  z2      ((xkoo+5*zab)+1-xleft,ykoo-3,f,p);
  mal     ((xkoo+6*zab)-1-xleft,ykoo, f,p);
  klein_d((xkoo+7*zab) -xleft,ykoo, f,p);
end;

OVERLAY procedure text_mg_m3 (xkoo,ykoo:integer; zab,f,p:byte);
{lokal zu draw_grafik}
begin
  klein_m(xkoo,      ykoo, f,p);  klein_g((xkoo+1*zab+1),ykoo,f,p);
  sstr    ((xkoo+2*zab),ykoo, f,p); klein_m((xkoo+3*zab ),ykoo,f,p);
  z3      ((xkoo+4*zab),ykoo-3,f,p);
end;

```

```

OVERLAY procedure text_cm_m3 (xkoo,ykoo: integer; zab,f,p:byte);
  {lokal zu draw_grafik}
begin
  klein_c( xkoo,      ykoo ,f,p); klein_m((xkoo+1*zab),ykoo, f,p);
  z3      ((xkoo+2*zab),ykoo-3,f,p); sstr  ((xkoo+3*zab),ykoo, f,p);
  klein_m((xkoo+4*zab),ykoo ,f,p); z3      ((xkoo+5*zab),ykoo-3,f,p);
end;

OVERLAY procedure text_ortho (xkoo,ykoo: integer; zab,f,p:byte);
  {lokal zu draw_grafik}
begin
  OO(xkoo      ,ykoo,f,p); RR  (xkoo+ 1*zab,ykoo,f,p);
  TT(xkoo+ 2*zab,ykoo,f,p); HH  (xkoo+ 3*zab,ykoo,f,p);
  OO(xkoo+ 4*zab,ykoo,f,p); trstr(xkoo+ 5*zab,ykoo,f,p);
end;

OVERLAY procedure phosphat_xxport (xkoo,ykoo: integer; zab,f,p:byte; ch:char);
  {lokal zu draw_grafik}
{ Parameter  xkoo: x-Koordinate Anfang links oben;
  ykoo: y-Koordinate dto.
  ch: Zu- oder Abfluss; Zu=z; Ab=a}

const aco{Pfeilfarbe} = 9;

OVERLAY procedure text_ortho (xkoo,ykoo: integer; zab,f,p:byte);
  {lokal zu phosphat_xxport}
begin
  OO(xkoo      ,ykoo,f,p); RR  (xkoo+ 1*zab,ykoo,f,p);
  TT(xkoo+ 2*zab,ykoo,f,p); HH  (xkoo+ 3*zab,ykoo,f,p);
  OO(xkoo+ 4*zab,ykoo,f,p); trstr(xkoo+ 5*zab,ykoo,f,p);
end;

OVERLAY procedure text_mg_m3 (xkoo,ykoo:integer; zab,f,p:byte);
  {lokal zu phosphat_xxport}
begin
  klein_m( xkoo,      ykoo ,f,p);  klein_g((xkoo+1*zab+1),ykoo,f,p);
  sstr  ((xkoo+2*zab),ykoo, f,p);  klein_m((xkoo+3*zab ),ykoo,f,p);
  z3      ((xkoo+4*zab),ykoo-3,f,p);
end;

begin {of procedure phosphat_xxport}
  text_ortho(xkoo,ykoo,zab,f,p);
  {write 'PHOSPHAT'}
  PP((xkoo+ 6*zab),ykoo,f,p); HH((xkoo+ 7*zab),ykoo,f,p);
  OO((xkoo+ 8*zab),ykoo,f,p); SS((xkoo+ 9*zab),ykoo,f,p);
  PP((xkoo+10*zab),ykoo,f,p); HH((xkoo+11*zab),ykoo,f,p);
  AA((xkoo+12*zab),ykoo,f,p); TT((xkoo+13*zab),ykoo,f,p);
  {write 'IM ..FLUSS'};
  II((xkoo+ 2*zab),(ykoo+8),f,p); MM((xkoo+ 3*zab),(ykoo+8),f,p);

```

```

case ch of
  'z': begin
    ZZ((xkoo+ 5*zab), (ykoo+8), f, p); UU((xkoo+ 6*zab), (ykoo+8), f, p);
  end;
  'a': begin
    AA((xkoo+ 5*zab), (ykoo+8), f, p); BB((xkoo+ 6*zab), (ykoo+8), f, p);
  end;
end;
FF((xkoo+ 7*zab), (ykoo+8), f, p); LL((xkoo+ 8*zab), (ykoo+8), f, p);
UU((xkoo+ 9*zab), (ykoo+8), f, p); SS((xkoo+10*zab), (ykoo+8), f, p);
SS((xkoo+11*zab), (ykoo+8), f, p);
text_mg_m3((xkoo+55), (ykoo+18), zab, f, p);
{Pfeil}
hline((xkoo-4), xkoo+100, ykoo-4, aco, 0);
hline((xkoo-4), xkoo+100, ykoo+26, aco, 0);
vline((xkoo-4), (ykoo-4), (ykoo+26), aco, 0);
line((xkoo+100), (ykoo -4), (xkoo+100+10), (ykoo- 4+15), aco, 0);
line((xkoo+100), (ykoo+26), (xkoo+100+10), (ykoo+26-15), aco, 0);
line((xkoo+101), (ykoo -4), (xkoo+100+11), (ykoo- 4+15), aco, 0);
line((xkoo+101), (ykoo+26), (xkoo+100+11), (ykoo+26-15), aco, 0);
end;

begin {of procedure draw_grafik}
  res_640x350;
  lines_43;
  text_bypass;

  {Definition der Farbenpalette}
  {
    0 schwarz      unver.      Background
    1 tiefblau     unver.      Hypolimnion
    2 dunkelgruen  transf. 18  giftgruen  O2-Uebersaet.
    3 dunkelcyan   transf. 27  cyan, sehr hell  Beschriftung
    4 dunkelrot    transf. 36  grell-rot    O2-Defizit
    5 dunkelviolet transf. 33  dunkles Lila  O2-Zehrung
    6 braun        unver.      Seebecken, Zoopl.
    7 dunkelweiß   transf. 37  pink      Kurve O2-def
    8 dunkelgrau   unver.
    9 Int.-blau    transf. 25  hellblau    Metalimnion, Pfeil
    10 Int.gruen   unver.      Algen
    11 Int.-cyan   unver.      Epil., Kurve O2-sat
    12 Int.-rot    unver.
    13 Int.-violett transf. 20  braun      Licht 2
    14 gelb        unver.      Licht 1
    15 Int.-weiß   unver.      Skalen, Phosphat
  }
  colorpalette(2,18);
  colorpalette(3,27);
  colorpalette(4,36);
  colorpalette(5,33);
  colorpalette(7,37);
  colorpalette(9,25); {alt: 9,43 helles hellblau}
  colorpalette(13,20);

  co3 := 3; {Beschriftung}
  co6 := 6; {Terrain, Seenbecken, Zooplankton}

```

```

col10 := 10; {Text Algen}
col11 := 11; {Text durchmischten Schicht}
col13 := 13; {Text Licht 2}
col14 := 14; {Text Licht 1}
col15 := 15; {Skalen, Maßeinheiten, Text Phosphat}

{draw Terrain}
for i := 1 to 2 do
begin
  hline(win[2,xol], (win[2,xur]+1), (win[2,yol]+1+i), co6,0);
  hline((win[4,xol]-1), win[4,xur], (win[4,yol]+1+i), co6,0);
end;

{draw Seebecken}
for i := 1 to (win[3,xur] - win[3,xol] - 1) do
begin
  pset((i+win[3,xol]), (win[3,yol]+baseline1[i]), co6,0);
  pset((i+win[3,xol]), (win[3,yol]+baseline2[i]), co6,0);
  pset((i+win[3,xol]), (win[3,yol]+baseline3[i]), co6,0);
end;

{draw Phosphat-Import}
phosphat_XXport((win[2,xol]+4), (win[2,yol]-29), 7, co3,0, 'z');
phosphat_XXport((win[4,xol]+6), (win[4,yol]-29), 7, co3,0, 'a');

{Window 4 Sauerstoff}
{draw y-Skala: Tiefe}
vline(win[4,xol], (win[4,yol]+4), win[4,yur], co15,0);
{Skalierung Tiefenskala -}
for i := 1 to 5 do
  hline((win[4,xol]-3), (win[4,xol]-1), (win[4,yol]-1+i*20), co15,0);
{Beschriftung Tiefenskala '0m, 5m, 10m, 15m, 20m, 25m'}
yanf := win[4,yol]-4; {y-Anf.}
xanf := win[4,xol]-20; {Zehnerstellen}
z1(xanf, (yanf+2*20), co15,0);
z1(xanf, (yanf+3*20), co15,0);
z2(xanf, (yanf+4*20), co15,0);
z2(xanf, (yanf+5*20), co15,0);
xanf := win[4,xol]-15; {Einerstellen}
z5(xanf, (yanf+1*20), co15,0);
OO(xanf, (yanf+2*20), co15,0);
z5(xanf, (yanf+3*20), co15,0);
OO(xanf, (yanf+4*20), co15,0);
z5(xanf, (yanf+5*20), co15,0);
xanf := win[4,xol]-10; {Meter}
for i := 1 to 5 do
  klein_m(xanf, (yanf+i*20), co15,0);
{write 'TIEFE '}
xanf := win[4,xol] - 32;
yanf := win[4,yol] + 36;
co := co3; ab := 10;
TT(xanf, yanf, co,0); II(xanf, (yanf+1*ab), co,0);
EE(xanf, yanf+2*ab, co,0); FF(xanf, (yanf+3*ab), co,0);
EE(xanf, yanf+4*ab, co,0); pfeilu(xanf, (yanf+5*ab), co,0);

```

```

{draw x-Skala oben und unten}
hline(win[4,xol],(win[4,xol]+(8*15-1)),(win[4,yol]+4),col5,0); {15mg a 8 pixel}
hline(win[4,xol],(win[4,xol]+(8*15-1)),win[4,yur],col5,0); {15mg a 8 pixel}
{Skalierung x-Skala oben und unten}
for i:=0 to 4 do
  vline((win[4,xol]+i*24),(win[4,yol]+4),(win[4,yol]+3),col5,0);
for i:=0 to 4 do
  vline((win[4,xol]+i*24),(win[4,yur]+1),(win[4,yur]+2),col5,0);
{Beschriftung x-Skala '3 6 9 12 mg/l'}
yanf := win[4,yur] + 5;
xanf := win[4,xol];
OO(xanf,yanf,col5,0);
z3(xanf+23,yanf,col5,0); z6((xanf+47),yanf,col5,0); z9((xanf+71),yanf,col5,0);
z1((xanf+91),yanf,col5,0); z2((xanf+96),yanf,col5,0);
klein_m((xanf+102),yanf,col5,0); klein_g((xanf+108),yanf,col5,0);
sstr((xanf+112),yanf,col5,0); klein_l((xanf+117),yanf,col5,0);
{Beschriftung 'SAUERSTOFFKONZENTRATION'}
yanf := win[4,yur] + 12;
xanf := win[4,xol]-8;
ab := 5; co := co3;
SS(xanf,yanf,co,0); AA((xanf+ 1*ab),yanf,co,0);
UU((xanf+ 2*ab),yanf,co,0); EE((xanf+ 3*ab),yanf,co,0);
RR((xanf+ 4*ab),yanf,co,0); SS((xanf+ 5*ab),yanf,co,0);
TT((xanf+ 6*ab),yanf,co,0); OO((xanf+ 7*ab),yanf,co,0);
FF((xanf+ 8*ab),yanf,co,0); FF((xanf+ 9*ab),yanf,co,0);
KK((xanf+10*ab),yanf,co,0); OO((xanf+11*ab),yanf,co,0);
NN((xanf+12*ab),yanf,co,0); ZZ((xanf+13*ab),yanf,co,0);
EE((xanf+14*ab),yanf,co,0); NN((xanf+15*ab),yanf,co,0);
TT((xanf+16*ab),yanf,co,0); RR((xanf+17*ab),yanf,co,0);
AA((xanf+18*ab),yanf,co,0); TT((xanf+19*ab),yanf,co,0);
II((xanf+20*ab),yanf,co,0); OO((xanf+21*ab),yanf,co,0);
NN((xanf+22*ab),yanf,co,0); Pfeilr((xanf+24*ab),yanf,co,0);

{draw Achsenkreuz Windows 5 - 8}
for i := 5 to 8 do
  begin
    {y-Achse}
    vline((win[i,xol]+XALI),(win[i,yur]-YAUN),(win[i,yol]+YAOB),col5,0);
    {Pfeilspitze y-Achse}
    line((win[i,xol]+XALI),(win[i,yol]+YAOB),
          (win[i,xol]+XALI-3),(win[i,yol]+YAOB+3),col5,0);
    line((win[i,xol]+XALI),(win[i,yol]+YAOB),
          (win[i,xol]+XALI+3),(win[i,yol]+YAOB+3),col5,0);
    {x-Achse}
    hline((win[i,xol]+XALI),(win[i,xur]-XARE),(win[i,yur]-YAUN),col5,0);
    {Skalierung x-Achse}
    for j := 0 to 12 do
      vline((j * 12 + (win[i,xol]+XALI+1)),
            (win[i,yur]-YAUN+1),(win[i,yur]-YAUN+3),col5,0);
    {Beschriftung x-Achse}
    yanf := win[i,yur]-YAUN + 2; {y-Position der Zeichen}
    ab := 12; {Zeichenabstand} co := col5;
    JJ((win[i,xol]+XALI+5),yanf,co,0);
    FF((win[i,xol]+XALI+5+ ab),yanf,co,0);
    MM((win[i,xol]+XALI+5+ 2*ab),yanf,co,0);
  end

```

```

AA((win[i,xol]+XALI+5+ 3*ab),yanf,co,0);
MM((win[i,xol]+XALI+5+ 4*ab),yanf,co,0);
JJ((win[i,xol]+XALI+5+ 5*ab),yanf,co,0);
JJ((win[i,xol]+XALI+5+ 6*ab),yanf,co,0);
AA((win[i,xol]+XALI+5+ 7*ab),yanf,co,0);
SS((win[i,xol]+XALI+5+ 8*ab),yanf,co,0);
OO((win[i,xol]+XALI+5+ 9*ab),yanf,co,0);
NN((win[i,xol]+XALI+5+10*ab),yanf,co,0);
DD((win[i,xol]+XALI+5+11*ab),yanf,co,0);
end;
{Skalierung y-Achse Window 5, Maßeinheit}
{y-Achse ist nur! 102 Punkte hoch; 1 J/cm2*d = 0.03 points; Max 3000 J}
xanf{Einerstelle bei CASE} := win[5,xol] + XALI - 14;
for i := 1 to 6 do {6 * '15 für jeweils 500 J'}
begin
hline(win[5,xol]+XALI-2,win[5,xol]+XALI+2,win[5,yur]-YAUN-i*15,co15,0);
case i of
1,3,5: z5(win[5,xol]+XALI-6,(win[5,yur]-YAUN-2)-i*15,co15,0);
2,4,6: OO(win[5,xol]+XALI-7,(win[5,yur]-YAUN-2)-i*15,co15,0);
end;
pkt(win[5,xol]+XALI-12,(win[5,yur]-YAUN-2)-i*15,co15,0);
case i of
1: OO(xanf,(win[5,yur]-YAUN-2)-i*15,co15,0);
2,3: z1(xanf,(win[5,yur]-YAUN-2)-i*15,co15,0);
4,5: z2(xanf,(win[5,yur]-YAUN-2)-i*15,co15,0);
6: z3(xanf,(win[5,yur]-YAUN-2)-i*15,co15,0);
end;
end;
text_kj(win[5,xol]+2,win[5,yol]+2,5,co15,0,true);
{Skalierung y-Achse Window 6, Maßeinheit}
{y-Achse ist 112 Punkte hoch; 1 mg P/m3 = 1.5 points; Max 70 mg}
xanf{Zehnerstelle bei CASE} := win[6,xol] + XALI - 12;
for i := 1 to 7 do {7 * '15 für jeweils 10 mg'}
begin
hline(win[6,xol]+XALI-2,win[6,xol]+XALI+2,win[6,yur]-YAUN-i*15,co15,0);
OO(win[6,xol]+XALI-7,(win[6,yur]-YAUN-2)-i*15,co15,0);
case i of
1: z1(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
2: z2(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
3: z3(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
4: z4(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
5: z5(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
6: z6(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
7: z7(xanf,(win[6,yur]-YAUN-2)-i*15,co15,0);
end;
end;
text_mg_m3(win[6,xol]+2,win[6,yol]+2,5,co15,0);
{Skalierung y-Achse Window 7, Maßeinheit}
{y-Achse ist 112 Punkte hoch; 1 cm3/m3 X = 5 points; Max 21 cm}
xanf{Zehnerstelle bei CASE} := win[7,xol] + XALI - 12;

```

```

for i := 1 to 7 do {7 * '15 für jeweils 3 cm'}
begin
  hline(win[7,xol]+XALI-2,win[7,xol]+XALI+2,win[7,yur]-YAUN-i*15,col5,0);
  case i of
    1: z3(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
    2: z6(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
    3: z9(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
    4: z2(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
    5: z5(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
    6: z8(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
    7: z1(win[7,xol]+XALI-7,(win[7,yur]-YAUN-2)-i*15,col5,0);
  end;
  case i of
    4..6: z1(xanf,(win[7,yur]-YAUN-2)-i*15,col5,0);
    7: z2(xanf,(win[6,yur]-YAUN-2)-i*15,col5,0);
  end;
end;
text_cm_m3(win[7,xol]+2,win[7,yol]+2,5,col5,0);
{Skalierung y-Achse Window 8, Maßeinheit}
{y-Achse ist nur! 102 Punkte hoch; 1 cm3/m3 Z = 8 points; Max 12 cm}
xanf{Zehnerstelle bei CASE} := win[8,xol] + XALI - 12;
for i := 1 to 6 do {6 * '16 für jeweils 3 cm'}
begin
  hline(win[8,xol]+XALI-2,win[8,xol]+XALI+2,win[8,yur]-YAUN-i*16,col5,0);
  case i of
    1: z2(win[8,xol]+XALI-7,(win[8,yur]-YAUN-2)-i*16,col5,0);
    2: z4(win[8,xol]+XALI-7,(win[8,yur]-YAUN-2)-i*16,col5,0);
    3: z6(win[8,xol]+XALI-7,(win[8,yur]-YAUN-2)-i*16,col5,0);
    4: z8(win[8,xol]+XALI-7,(win[8,yur]-YAUN-2)-i*16,col5,0);
    5: O0(win[8,xol]+XALI-7,(win[8,yur]-YAUN-2)-i*16,col5,0);
    6: z2(win[8,xol]+XALI-7,(win[8,yur]-YAUN-2)-i*16,col5,0);
  end;
  case i of
    5..6: z1(xanf,(win[8,yur]-YAUN-2)-i*16,col5,0);
  end;
end;
text_cm_m3(win[8,xol]+2,win[8,yol]+2,5,col5,0);

{Text 'IN DER' Window 6 - 8 (*7*)}
co := co3; ab := 5;{Zeichenabstand}
for i := 6 to 8 {7} do
begin
  xanf := win[i,xol]+123; yanf := win[i,yur]-27;
  II(xanf,yanf,co,0); NN(xanf+ 1*ab,yanf,co,0);
  DD(xanf+ 3*ab,yanf,co,0); EE(xanf+ 4*ab,yanf,co,0);
  RR(xanf+ 5*ab,yanf,co,0);
end;

{Text 'DURCHMISCHTEN SCHICHT' Window 6 - 8}
co := co11; ab := 6;{Zeichenabstand}
for i := 6 to 8 do
begin
  xanf := win[i,xol]+26; yanf := win[i,yur]-20;
  DD(xanf,yanf,co,0); UU(xanf+ 1*ab,yanf,co,0);
  RR(xanf+ 2*ab,yanf,co,0); CC(xanf+ 3*ab,yanf,co,0);
end;

```

```

HH(xanf+ 4*ab,yanf,co,0); MM(xanf+ 5*ab,yanf,co,0);
II(xanf+ 6*ab,yanf,co,0); SS(xanf+ 7*ab,yanf,co,0);
CC(xanf+ 8*ab,yanf,co,0); HH(xanf+ 9*ab,yanf,co,0);
TT(xanf+10*ab,yanf,co,0); EE(xanf+11*ab,yanf,co,0);
NN(xanf+12*ab,yanf,co,0); SS(xanf+14*ab,yanf,co,0);
CC(xanf+15*ab,yanf,co,0); HH(xanf+16*ab,yanf,co,0);
II(xanf+17*ab,yanf,co,0); CC(xanf+18*ab,yanf,co,0);
HH(xanf+19*ab,yanf,co,0); TT(xanf+20*ab,yanf,co,0);
end;

```

```

{Text 'LICHT'}
e_gotoxy(9,41,0); writepage(0); textattr(col4,0);
write('LICHT');
{write 'UBER DER W.O.'}
xanf := win[5,xol]; yanf := win[6,yur]-19; ab := 5; co := col4;
UE(xanf      ,yanf,co,0); BB(xanf+ 1*ab,yanf,co,0);
EE(xanf+ 2*ab,yanf,co,0); RR(xanf+ 3*ab,yanf,co,0);
DD(xanf+ 5*ab,yanf,co,0); EE(xanf+ 6*ab,yanf,co,0);
RR(xanf+ 7*ab,yanf,co,0); WW(xanf+ 9*ab,yanf,co,0);
pkt(xanf+10*ab,yanf,co,0); OO(xanf+11*ab,yanf,co,0);
pkt(xanf+12*ab,yanf,co,0);
text_kj(xanf+119,yanf,6,col5,0,false);
{write 'UNTER DER W.O.'}
xanf := win[5,xol]; yanf := win[6,yur]-11; ab := 5; co := col3;
z3(xanf      ,yanf,co,0); klein_m(xanf+ 1*ab,yanf,co,0);
UU(xanf+ 3*ab,yanf,co,0); NN(xanf+ 4*ab,yanf,co,0);
TT(xanf+ 5*ab,yanf,co,0); EE(xanf+ 6*ab,yanf,co,0);
RR(xanf+ 7*ab,yanf,co,0);
WW(xanf+ 9*ab,yanf,co,0); pkt(xanf+10*ab,yanf,co,0);
OO(xanf+11*ab,yanf,co,0); pkt(xanf+12*ab,yanf,co,0);
text_kj(xanf+119,yanf,6,col5,0,false);

{Text 'ORTHO-PHOSPHAT'}
xanf := win[6,xol]+20; yanf := win[6,yur]-27;
text_ortho(xanf,yanf,6,co3,0);
e_gotoxy(28,41,0); writepage(0); textattr(col5,0); write('PHOSPHAT');
text_mg_m3(xanf+90,yanf+16,6,col5,0);

```

```

{Text '(*KIESEL-*)ALGEN'}
xanf := win[7,xol]+19; yanf := win[7,yur]-27; ab := 6; co := co3;
KK(xanf      ,yanf,co,0); II(xanf+ 1*ab,yanf,co,0);
EE(xanf+ 2*ab,yanf,co,0); SS(xanf+ 3*ab,yanf,co,0);
EE(xanf+ 4*ab,yanf,co,0); LL(xanf+ 5*ab,yanf,co,0);
trstr(xanf+ 6*ab,yanf,co,0);

```

```

e_gotoxy(49,41,0); writepage(0); textattr(col10,0); write('ALGEN');
text_cm_m3(xanf+94,yanf+16,6,col5,0);

```

```

{Text '(*DAPHNIEN-*)ZOOPLANKTON'}
xanf := win[8,xol]+11; yanf := win[8,yur]-27; ab := 5; co := co3;
DD(xanf      ,yanf,co,0); AA(xanf+ 1*ab,yanf,co,0);
PP(xanf+ 2*ab,yanf,co,0); HH(xanf+ 3*ab,yanf,co,0);
NN(xanf+ 4*ab,yanf,co,0); II(xanf+ 5*ab,yanf,co,0);
EE(xanf+ 6*ab,yanf,co,0); NN(xanf+ 7*ab,yanf,co,0);
trstr(xanf+ 8*ab,yanf,co,0);

```

```

}
  e_gotoxy({68}65,41,0); writepage(0); textattr(co6,0); write('ZOOPLANKTON');
{
  xanf := win[i,xol]+145; yanf := win[i,yur]-27;
  II(xanf ,yanf,co,0); pkt(xanf+ 1*ab-2,yanf,co,0);
  DD(xanf+ 2*ab-2,yanf,co,0); pkt(xanf+ 3*ab-4,yanf,co,0);
}
  text_cm_m3(xanf+105,yanf+16,6,col5,0);
end; {of procedure draw_grafik}

{----- Bestaetigung der Wahl 'Abbruch' -----}
OVERLAY procedure abbruch_bestatigung;
var zeichen: char;

begin
  text_80x25;
  text_bypass;
  writepage(0);
  textattr(14,0);
  e_gotoxy(21,12,0); write('SIMULATION BEENDEN ..... J (für JA)');
  e_gotoxy(40,14,0); write('..... N (für NEIN)');
  e_gotoxy(30,18,0); write('IHRE AUSWAHL:');
  zeichen := ' ';
  flush_keyboard;
  e_gotoxy(44,18,0);
  blink_cursor(44,18,10);
  read(Kbd,zeichen);
  textattr(10,0); write(zeichen);
  delay(350);
  flush_keyboard;
  zeichen := UpCase(zeichen);
  if (zeichen = 'J') or (zeichen = 'N')
  then
    case zeichen of
      'J': flag_abbruch := true;
      'N': exit;
    end
  else abbruch_bestatigung;
end;

{----- Auswahl der Fortsetzung -----}
OVERLAY procedure menu_weiter;
var zeichen: char;

begin
  text_80x25;
  text_bypass;
  writepage(0);
  textattr(14,0);
  if flag_last = false
  then begin
    e_gotoxy(21,7,0); write('SIMULATION FORTSETZEN ..... 1');
    e_gotoxy(22,8,0); write('(nächstes Modelljahr)');
    end;
  e_gotoxy(21,11,0); write('SIMULATION WIEDERHOLEN ..... 2');

```

```

e_gotoxy(22,12,0); write('gleiches Modelljahr,');
e_gotoxy(23,13,0); write('andere Parameter');
e_gotoxy(21,15,0); write('SIMULATION BEENDEN ..... 3');
e_gotoxy(30,19,0); write('IHRE AUSWAHL:');
zeichen := ' ';
flush_keyboard;
e_gotoxy(44,19,0);
blink_cursor(44,19,10);
read(Kbd,zeichen);
textattr(10,0); write(zeichen);
delay(350);
flush_keyboard;
if flag_last = false
then begin
    if zeichen in ['1'..'3']
    then
        case zeichen of
            '1':exit;
            '2':begin
                cumsed := cumsedanf;
                jahr := jahr - 1;
                exit;
            end;
            '3':abbruch_bestatigung;
        end
    else menu_weiter;
    end
else begin
    if zeichen in ['2'..'3']
    then
        case zeichen of
            '2':begin
                cumsed := cumsedanf;
                jahr := jahr - 1;
                exit;
            end;
            '3':abbruch_bestatigung;
        end
    else menu_weiter;
    end
end;

{----- Auswahl der zu veraendernden Parameter -----}
OVERLAY procedure experimente;

type asctype = string[15];
var neustring: string[80];

```

```

OVERLAY procedure text_experimente; {lokal zu experimente}
begin
  text_80x25;
  text_bypass;
  writepage(0);
  textattr(10,0);
  e_gotoxy( 1, 1,0); write('Folgende PARAMETER können zu Experimentierzwecken');
  e_gotoxy(51, 1,0); write('verändert werden:');
  textattr(15,0);
  e_gotoxy( 1, 3,0); write('Optimaltemperatur');
  e_gotoxy( 1, 4,0); write('      für Algenwachstum [°C] ..... ');
  e_gotoxy( 1, 5,0); write('      für Zooplanktonfraß [°C] .... ');
  e_gotoxy( 1, 7,0); write('Phosphatausnutzungskoeffizient');
  e_gotoxy( 1, 8,0); write('      für Algen [cm3/mg P] ..... ');
  e_gotoxy( 1, 9,0); write('      für Zooplankton [cm3/mg P] .. ');
  e_gotoxy( 1,11,0); write('Faktor für die physiologische');
  e_gotoxy( 1,12,0); write('      Verwertbarkeit der Algen');
  e_gotoxy( 1,13,0); write('      durch das Zooplankton ..... ');
  e_gotoxy( 1,15,0); write('Lichthemmungsfaktor');
  e_gotoxy( 1,16,0); write('      für Zooplanktonfraß ..... ');
  e_gotoxy( 1,18,0); write('Nettosinkgeschwindigkeit');
  e_gotoxy( 1,19,0); write('      der Algen [m/d] ..... ');
  textattr(11,0);
  e_gotoxy(40,4,0); write(xgtopt:4:1);
  e_gotoxy(40,5,0); write(zgztopt:4:1);
  e_gotoxy(40,8,0); write(kxmup:3:1);
  e_gotoxy(40,9,0); write(kznup:3:1);
  e_gotoxy(40,13,0); write(uxz:3:1);
  e_gotoxy(40,16,0); write(zgzi:3:1);
  e_gotoxy(40,19,0); write(xs:3:1);
end;

```

```

procedure eingabefeld2(xpos,ypos,co: byte; bereich: asctype;
                       blink: boolean); {lokal zu experimente}
begin
  e_gotoxy(xpos,ypos,0); textattr(co,0);
  write('neuer Wert (' ,bereich, '): ',neusttring);
  e_gotoxy(xpos+26,ypos,0);
  if blink = true then blink_cursor(xpos+26,ypos,14);
end;

```

```

OVERLAY procedure aendern; {lokal zu experimente}
var i:
  fehler:
  pruefchar:
  xgtoptneu, zgztoptneu, kxmupneu, kznupneu,
  wxzneu, zgzineu, xsneu:
byte;
integer;
char;
real;

```

```
OVERLAY procedure tempzneu; {lokal zu aendern}
```

```
begin
  neustring := '          ';
  eingabefeld2(46,4,10,'0.1...25.0',true);
  textattr(10,0);
  readln(neustring);
  fehler := 0;
  if (length(neustring) <= 4) and (length(neustring) > 0)
  then begin
    for i := 1 to length(neustring) do
    begin
      pruefchar := copy(neustring,i,1);
      if not (pruefchar in ['0'..'9','.']) then fehler := 1;
    end;
    if fehler = 0
    then begin
      val(neustring,xgtoptneu,fehler);
      if (xgtoptneu < 0.1) or (xgtoptneu > 25.0) then fehler := 1;
      if fehler = 0
      then begin
        xgtopt := xgtoptneu;
        exit;
      end;
    end;
  end;
  write(chr(7));
  tempzneu;
end;
```

```
OVERLAY procedure tempzneu; {lokal zu aendern}
```

```
begin
  neustring := '          ';
  eingabefeld2(46,5,10,'0.1...25.0',true);
  textattr(10,0);
  readln(neustring);
  fehler := 0;
  if (length(neustring) <= 4) and (length(neustring) > 0)
  then begin
    for i := 1 to length(neustring) do
    begin
      pruefchar := copy(neustring,i,1);
      if not (pruefchar in ['0'..'9','.']) then fehler := 1;
    end;
    if fehler = 0
    then begin
      val(neustring,zgztptneu,fehler);
      if (zgztptneu < 0.1) or (zgztptneu > 25.0)
      then fehler := 1;
      if fehler = 0
      then begin
        zgztpt := zgztptneu;
        exit;
      end;
    end;
  end;
end;
```

```

write(Chr(7));
tempzneu;
end;

OVERLAY procedure pzkoeffneu; {lokal zu aendern}
begin
  neustring := '          ';
  eingabefeld2(46,8,10,'0.1...1.0',true);
  textattr(10,0);
  readln(neustring);
  fehler := 0;
  if (length(neustring) <= 3) and (length(neustring) > 0)
    then begin
      for i := 1 to length(neustring) do
        begin
          pruefchar := copy(neustring,i,1);
          if not (pruefchar in ['0'..'9','.']) then fehler := 1;
        end;
      if fehler = 0
        then begin
          val(neustring,komupneu,fehler);
          if (komupneu < 0.1) or (komupneu > 1.0) then fehler := 1;
          if fehler = 0
            then begin
              komup := komupneu;
              exit;
            end;
        end;
    end;
  write(Chr(7));
  pzkoeffneu;
end;

```

```

OVERLAY procedure pzkoeffneu; {lokal zu aendern}
begin
  neustring := '          ';
  eingabefeld2(46,9,10,'0.1...1.0',true);
  textattr(10,0);
  readln(neustring);
  fehler := 0;
  if (length(neustring) <= 3) and (length(neustring) > 0)
    then begin
      for i := 1 to length(neustring) do
        begin
          pruefchar := copy(neustring,i,1);
          if not (pruefchar in ['0'..'9','.']) then fehler := 1;
        end;
    end;

```

```

    if fehler = 0
    then begin
        val(neuststring,kznupneu,fehler);
        if (kznupneu < 0.1) or (kznupneu > 1.0) then fehler := 1;
        if fehler = 0
        then begin
            kznup := kznupneu;
            exit;
        end;
    end;
end;

write(Chr(7));
pzkoeffneu;
end;

```

```

OVERLAY procedure faktphys_neu; {lokal zu aendern}
begin
    neuststring := '          ';
    eingabefeld2(46,13,10,'0.1...1.0',true);
    textattr(10,0);
    readln(neuststring);
    fehler := 0;
    if (length(neuststring) <= 3) and (length(neuststring) > 0)
    then begin
        for i := 1 to length(neuststring) do
        begin
            pruefchar := copy(neuststring,i,1);
            if not (pruefchar in ['0'..'9','.']) then fehler := 1;
        end;
        if fehler = 0
        then begin
            val(neuststring,uxzneu,fehler);
            if (uxzneu < 0.1) or (uxzneu > 1.0) then fehler := 1;
            if fehler = 0
            then begin
                uxz := uxzneu;
                exit;
            end;
        end;
    end;
    write(Chr(7));
    faktphys_neu;
end;

```

```

OVERLAY procedure lichthem_neu; {lokal zu aendern}
begin
    neuststring := '          ';
    eingabefeld2(46,16,10,'0.1...0.5',true);
    textattr(10,0);
    readln(neuststring);
    fehler := 0;

```

```

if (length(neustring) <= 3) and (length(neustring) > 0)
  then begin
    for i := 1 to length(neustring) do
      begin
        pruefchar := copy(neustring,i,1);
        if not (pruefchar in ['0'..'9','.']) then fehler := 1;
      end;
    if fehler = 0
      then begin
        val(neustring,zgzineu,fehler);
        if (zgzineu < 0.1) or (zgzineu > 0.5) then fehler := 1;
        if fehler = 0
          then begin
            zgzi := zgzineu;
            exit;
          end;
        end;
      end;
    write(Chr(7));
    lichthemm_neu;
  end;

OVERLAY procedure vxsed_neu; {lokal zu aendern}
begin
  neustring := '          ';
  eingabefeld2(46,19,10,'0.05...1.0',true);
  textattr(10,0);
  readln(neustring);
  fehler := 0;
  if (length(neustring) <= 4) and (length(neustring) > 0)
    then begin
      for i := 1 to length(neustring) do
        begin
          pruefchar := copy(neustring,i,1);
          if not (pruefchar in ['0'..'9','.']) then fehler := 1;
        end;
      if fehler = 0
        then begin
          val(neustring,xsneu,fehler);
          if (xsneu < 0.05) or (xsneu > 1.0) then fehler := 1;
          if fehler = 0
            then begin
              xs := xsneu;
              exit;
            end;
          end;
        end;
      end;
    write(Chr(7));
    vxsed_neu;
  end;

```

```

begin (of aendern)
  tempneu;
  eingabefeld2(46,4,0,'0.1...20.0',false); e_gotoxy(72,4,0); write(' ');
  e_gotoxy(40,4,0); textattr(11,0); write(xgtopt:4:1);
  tempzneu;
  eingabefeld2(46,5,0,'0.1...20.0',false); e_gotoxy(72,5,0); write(' ');
  e_gotoxy(40,5,0); textattr(11,0); write(zgzt opt:4:1);
  pxkoeffneu;
  eingabefeld2(46,8,0,'0.1...1.0',false); e_gotoxy(72,8,0); write(' ');
  e_gotoxy(40,8,0); textattr(11,0); write(kznup:4:1);
  pzkoefneu;
  eingabefeld2(46,9,0,'0.1...1.0',false); e_gotoxy(72,9,0); write(' ');
  e_gotoxy(40,9,0); textattr(11,0); write(kznup:4:1);
  faktphys_neu;
  eingabefeld2(46,13,0,'0.1...1.0',false); e_gotoxy(72,13,0); write(' ');
  e_gotoxy(40,13,0); textattr(11,0); write(uxz:4:1);
  lichthem_neu;
  eingabefeld2(46,16,0,'0.1...0.5',false); e_gotoxy(72,16,0); write(' ');
  e_gotoxy(40,16,0); textattr(11,0); write(zgzi:4:1);
  vxsed_neu;
  eingabefeld2(46,19,0,'0.05...0.3',false); e_gotoxy(72,19,0); write(' ');
  e_gotoxy(40,19,0); textattr(11,0); write(xs:4:1);
end;

```

```

procedure bypass_verzweigung; forward;
OVERLAY procedure verzweigung; {lokal zu experimente}
var zeichen : char;

```

```

begin
  e_gotoxy(21,20,0); textattr(14,0); write(' ');
  e_gotoxy(21,21,0); write(' Simulation fortsetzen ..... 1 ');
  e_gotoxy(21,22,0); write(' Daten ändern ..... 2 ');
  e_gotoxy(21,23,0); write(' ');
  e_gotoxy(21,24,0); write(' Ihre Eingabe: ');
  e_gotoxy(21,25,0); write(' ');

```

```

  zeichen := ' ';
  flush_keyboard;
  e_gotoxy(48,24,0);
  blink_cursor(48,24,10);
  read(Kbd,zeichen);
  textattr(10,0); write(zeichen);
  delay(350);
  flush_keyboard;
  if zeichen in ['1'..'2']
  then
    case zeichen of
      '1': exit;
      '2': begin
        aendern;
        bypass_verzweigung;
      end;
    end;
  end;
end

```

```

        else begin
            write(Chr(7));
            bypass_verzweigung;
        end;
end;

procedure bypass_verzweigung;
begin
    verzweigung;
end;

begin {of procedure experimente}
    text_experimente;
    bypass_verzweigung;
    exit;
end;

{----- Auswahl des Simulationslaufs -----}
OVERLAY procedure menu_laufotyp;
var zeichen: char;

begin
    text_80x25;
    text_bypass;
    writepage(0);
    textattr(11,0);
    e_gotoxy(21, 9,0); write('SIMULATION:  NORMALLAUF ..... 1');
    textattr(15,0);
    e_gotoxy(21,12,0); write('SIMULATION:  EXPERIMENTE ..... 2');
    e_gotoxy(21,13,0); write(' (Es können einige zusätzliche)');
    e_gotoxy(21,14,0); write(' Parameter verändert werden. ');
    textattr(14,0); e_gotoxy(30,19,0); write('IHRE AUSWAHL:');
    zeichen := ' ';
    flush_keyboard;
    e_gotoxy(44,19,0);
    blink_cursor(44,19,10);
    read(Kbd,zeichen);
    textattr(10,0); write(zeichen);
    delay(350);
    flush_keyboard;
    if zeichen in ['1'..'2']
    then
        case zeichen of
            '1': exit;
            '2': begin
                    experimente;
                    exit;
                end;
        end
    else menu_laufotyp;
end;

```

 TEMPVERT.OVL

Diese Datei enthält die Tabellen für die vertikale Temperaturverteilung.

```
{type jz = (j0,j5,j10,j15,j20,j25); var tj: array[jz,1..36] of integer;}
```

```
OVERLAY procedure temperaturverteilung_1; {warmes Jahr}
```

```
begin
```

```
  tj[j0, 1] := 3;  tj[j0, 2] := 1;  tj[j0, 3] := 0;  tj[j0, 4] := 0;
  tj[j0, 5] := 0;  tj[j0, 6] := 1;  tj[j0, 7] := 2;  tj[j0, 8] := 4;
  tj[j0, 9] := 5;  tj[j0,10] := 7;  tj[j0,11] := 9;  tj[j0,12] :=10;
  tj[j0,13] :=12;  tj[j0,14] :=14;  tj[j0,15] :=16;  tj[j0,16] :=18;
  tj[j0,17] :=19;  tj[j0,18] :=20;  tj[j0,19] :=21;  tj[j0,20] :=22;
  tj[j0,21] :=23;  tj[j0,22] :=24;  tj[j0,23] :=24;  tj[j0,24] :=23;
  tj[j0,25] :=22;  tj[j0,26] :=21;  tj[j0,27] :=20;  tj[j0,28] :=19;
  tj[j0,29] :=17;  tj[j0,30] :=15;  tj[j0,31] :=13;  tj[j0,32] :=11;
  tj[j0,33] := 8;  tj[j0,34] := 5;  tj[j0,35] := 4;  tj[j0,36] := 4;
```

```
  tj[j5, 1] := 4;  tj[j5, 2] := 3;  tj[j5, 3] := 2;  tj[j5, 4] := 2;
  tj[j5, 5] := 2;  tj[j5, 6] := 2;  tj[j5, 7] := 2;  tj[j5, 8] := 4;
  tj[j5, 9] := 4;  tj[j5,10] := 5;  tj[j5,11] := 6;  tj[j5,12] := 8;
  tj[j5,13] :=10;  tj[j5,14] :=12;  tj[j5,15] :=14;  tj[j5,16] :=16;
  tj[j5,17] :=18;  tj[j5,18] :=19;  tj[j5,19] :=20;  tj[j5,20] :=20;
  tj[j5,21] :=21;  tj[j5,22] :=22;  tj[j5,23] :=23;  tj[j5,24] :=23;
  tj[j5,25] :=22;  tj[j5,26] :=21;  tj[j5,27] :=20;  tj[j5,28] :=19;
  tj[j5,29] :=17;  tj[j5,30] :=15;  tj[j5,31] :=13;  tj[j5,32] :=11;
  tj[j5,33] := 8;  tj[j5,34] := 5;  tj[j5,35] := 4;  tj[j5,36] := 4;
```

```
  tj[j10, 1] := 4;  tj[j10, 2] := 4;  tj[j10, 3] := 3;  tj[j10, 4] := 3;
  tj[j10, 5] := 3;  tj[j10, 6] := 3;  tj[j10, 7] := 3;  tj[j10, 8] := 4;
  tj[j10, 9] := 4;  tj[j10,10] := 4;  tj[j10,11] := 5;  tj[j10,12] := 6;
  tj[j10,13] := 8;  tj[j10,14] := 9;  tj[j10,15] :=10;  tj[j10,16] :=11;
  tj[j10,17] :=12;  tj[j10,18] :=13;  tj[j10,19] :=14;  tj[j10,20] :=14;
  tj[j10,21] :=15;  tj[j10,22] :=15;  tj[j10,23] :=16;  tj[j10,24] :=16;
  tj[j10,25] :=16;  tj[j10,26] :=15;  tj[j10,27] :=15;  tj[j10,28] :=15;
  tj[j10,29] :=15;  tj[j10,30] :=15;  tj[j10,31] :=13;  tj[j10,32] :=11;
  tj[j10,33] := 8;  tj[j10,34] := 5;  tj[j10,35] := 4;  tj[j10,36] := 4;
```

```
  tj[j15, 1] := 4;  tj[j15, 2] := 4;  tj[j15, 3] := 3;  tj[j15, 4] := 3;
  tj[j15, 5] := 3;  tj[j15, 6] := 3;  tj[j15, 7] := 4;  tj[j15, 8] := 4;
  tj[j15, 9] := 4;  tj[j15,10] := 4;  tj[j15,11] := 4;  tj[j15,12] := 5;
  tj[j15,13] := 6;  tj[j15,14] := 7;  tj[j15,15] := 8;  tj[j15,16] := 9;
  tj[j15,17] :=10;  tj[j15,18] :=10;  tj[j15,19] :=10;  tj[j15,20] :=10;
  tj[j15,21] :=11;  tj[j15,22] :=11;  tj[j15,23] :=12;  tj[j15,24] :=12;
  tj[j15,25] :=12;  tj[j15,26] :=11;  tj[j15,27] :=11;  tj[j15,28] :=11;
  tj[j15,29] :=10;  tj[j15,30] :=10;  tj[j15,31] :=10;  tj[j15,32] :=10;
  tj[j15,33] := 8;  tj[j15,34] := 5;  tj[j15,35] := 4;  tj[j15,36] := 4;
```

```

tj[j20, 1] := 4;  tj[j20, 2] := 4;  tj[j20, 3] := 4;  tj[j20, 4] := 4;
tj[j20, 5] := 4;  tj[j20, 6] := 4;  tj[j20, 7] := 4;  tj[j20, 8] := 4;
tj[j20, 9] := 4;  tj[j20,10] := 4;  tj[j20,11] := 4;  tj[j20,12] := 4;
tj[j20,13] := 5;  tj[j20,14] := 5;  tj[j20,15] := 6;  tj[j20,16] := 6;
tj[j20,17] := 6;  tj[j20,18] := 6;  tj[j20,19] := 6;  tj[j20,20] := 6;
tj[j20,21] := 7;  tj[j20,22] := 7;  tj[j20,23] := 7;  tj[j20,24] := 7;
tj[j20,25] := 7;  tj[j20,26] := 6;  tj[j20,27] := 6;  tj[j20,28] := 6;
tj[j20,29] := 6;  tj[j20,30] := 6;  tj[j20,31] := 6;  tj[j20,32] := 6;
tj[j20,33] := 6;  tj[j20,34] := 5;  tj[j20,35] := 4;  tj[j20,36] := 4;

tj[j25, 1] := 4;  tj[j25, 2] := 4;  tj[j25, 3] := 4;  tj[j25, 4] := 4;
tj[j25, 5] := 4;  tj[j25, 6] := 4;  tj[j25, 7] := 4;  tj[j25, 8] := 4;
tj[j25, 9] := 4;  tj[j25,10] := 4;  tj[j25,11] := 4;  tj[j25,12] := 4;
tj[j25,13] := 5;  tj[j25,14] := 5;  tj[j25,15] := 5;  tj[j25,16] := 5;
tj[j25,17] := 5;  tj[j25,18] := 5;  tj[j25,19] := 5;  tj[j25,20] := 5;
tj[j25,21] := 5;  tj[j25,22] := 5;  tj[j25,23] := 6;  tj[j25,24] := 6;
tj[j25,25] := 6;  tj[j25,26] := 5;  tj[j25,27] := 5;  tj[j25,28] := 5;
tj[j25,29] := 5;  tj[j25,30] := 5;  tj[j25,31] := 5;  tj[j25,32] := 5;
tj[j25,33] := 5;  tj[j25,34] := 5;  tj[j25,35] := 4;  tj[j25,36] := 4;
end;
```

```

OVERLAY procedure temperaturverteilung_2; {normales Jahr}
begin
```

```

  tj[ j0, 1] := 2;  tj[ j0, 2] := 1;  tj[ j0, 3] := 0;  tj[ j0, 4] := 0;
  tj[ j0, 5] := 0;  tj[ j0, 6] := 0;  tj[ j0, 7] := 2;  tj[ j0, 8] := 4;
  tj[ j0, 9] := 4;  tj[ j0,10] := 6;  tj[ j0,11] := 7;  tj[ j0,12] := 8;
  tj[ j0,13] :=10;  tj[ j0,14] :=12;  tj[ j0,15] :=14;  tj[ j0,16] :=15;
  tj[ j0,17] :=16;  tj[ j0,18] :=18;  tj[ j0,19] :=19;  tj[ j0,20] :=20;
  tj[ j0,21] :=21;  tj[ j0,22] :=22;  tj[ j0,23] :=21;  tj[ j0,24] :=20;
  tj[ j0,25] :=19;  tj[ j0,26] :=18;  tj[ j0,27] :=17;  tj[ j0,28] :=16;
  tj[ j0,29] :=14;  tj[ j0,30] :=12;  tj[ j0,31] :=11;  tj[ j0,32] := 9;
  tj[ j0,33] := 7;  tj[ j0,34] := 5;  tj[ j0,35] := 4;  tj[ j0,36] := 4;

  tj[ j5, 1] := 3;  tj[ j5, 2] := 3;  tj[ j5, 3] := 2;  tj[ j5, 4] := 2;
  tj[ j5, 5] := 2;  tj[ j5, 6] := 2;  tj[ j5, 7] := 2;  tj[ j5, 8] := 4;
  tj[ j5, 9] := 4;  tj[ j5,10] := 5;  tj[ j5,11] := 5;  tj[ j5,12] := 6;
  tj[ j5,13] := 8;  tj[ j5,14] :=10;  tj[ j5,15] :=12;  tj[ j5,16] :=13;
  tj[ j5,17] :=15;  tj[ j5,18] :=17;  tj[ j5,19] :=18;  tj[ j5,20] :=19;
  tj[ j5,21] :=20;  tj[ j5,22] :=20;  tj[ j5,23] :=20;  tj[ j5,24] :=20;
  tj[ j5,25] :=19;  tj[ j5,26] :=18;  tj[ j5,27] :=17;  tj[ j5,28] :=16;
  tj[ j5,29] :=14;  tj[ j5,30] :=12;  tj[ j5,31] :=11;  tj[ j5,32] := 9;
  tj[ j5,33] := 7;  tj[ j5,34] := 5;  tj[ j5,35] := 4;  tj[ j5,36] := 4;

  tj[j10, 1] := 4;  tj[j10, 2] := 4;  tj[j10, 3] := 3;  tj[j10, 4] := 3;
  tj[j10, 5] := 3;  tj[j10, 6] := 3;  tj[j10, 7] := 3;  tj[j10, 8] := 4;
  tj[j10, 9] := 4;  tj[j10,10] := 4;  tj[j10,11] := 4;  tj[j10,12] := 5;
  tj[j10,13] := 6;  tj[j10,14] := 8;  tj[j10,15] := 9;  tj[j10,16] :=10;
  tj[j10,17] :=11;  tj[j10,18] :=12;  tj[j10,19] :=13;  tj[j10,20] :=14;
  tj[j10,21] :=14;  tj[j10,22] :=14;  tj[j10,23] :=14;  tj[j10,24] :=14;
  tj[j10,25] :=14;  tj[j10,26] :=14;  tj[j10,27] :=14;  tj[j10,28] :=14;
  tj[j10,29] :=14;  tj[j10,30] :=12;  tj[j10,31] :=11;  tj[j10,32] := 9;
  tj[j10,33] := 7;  tj[j10,34] := 5;  tj[j10,35] := 4;  tj[j10,36] := 4;
```

```

tj[j15, 1] := 4;   tj[j15, 2] := 4;   tj[j15, 3] := 3;   tj[j15, 4] := 3;
tj[j15, 5] := 3;   tj[j15, 6] := 3;   tj[j15, 7] := 4;   tj[j15, 8] := 4;
tj[j15, 9] := 4;   tj[j15,10] := 4;   tj[j15,11] := 4;   tj[j15,12] := 4;
tj[j15,13] := 5;   tj[j15,14] := 6;   tj[j15,15] := 7;   tj[j15,16] := 8;
tj[j15,17] := 9;   tj[j15,18] :=10;   tj[j15,19] :=10;   tj[j15,20] :=10;
tj[j15,21] :=10;   tj[j15,22] :=10;   tj[j15,23] :=10;   tj[j15,24] :=10;
tj[j15,25] :=10;   tj[j15,26] :=10;   tj[j15,27] :=10;   tj[j15,28] :=10;
tj[j15,29] :=10;   tj[j15,30] :=10;   tj[j15,31] :=10;   tj[j15,32] := 9;
tj[j15,33] := 7;   tj[j15,34] := 5;   tj[j15,35] := 4;   tj[j15,36] := 4;

```

```

tj[j20, 1] := 4;   tj[j20, 2] := 4;   tj[j20, 3] := 4;   tj[j20, 4] := 4;
tj[j20, 5] := 4;   tj[j20, 6] := 4;   tj[j20, 7] := 4;   tj[j20, 8] := 4;
tj[j20, 9] := 4;   tj[j20,10] := 4;   tj[j20,11] := 4;   tj[j20,12] := 4;
tj[j20,13] := 4;   tj[j20,14] := 5;   tj[j20,15] := 5;   tj[j20,16] := 6;
tj[j20,17] := 6;   tj[j20,18] := 6;   tj[j20,19] := 6;   tj[j20,20] := 6;
tj[j20,21] := 6;   tj[j20,22] := 6;   tj[j20,23] := 6;   tj[j20,24] := 6;
tj[j20,25] := 6;   tj[j20,26] := 6;   tj[j20,27] := 6;   tj[j20,28] := 6;
tj[j20,29] := 6;   tj[j20,30] := 6;   tj[j20,31] := 6;   tj[j20,32] := 6;
tj[j20,33] := 6;   tj[j20,34] := 5;   tj[j20,35] := 4;   tj[j20,36] := 4;

```

```

tj[j25, 1] := 4;   tj[j25, 2] := 4;   tj[j25, 3] := 4;   tj[j25, 4] := 4;
tj[j25, 5] := 4;   tj[j25, 6] := 4;   tj[j25, 7] := 4;   tj[j25, 8] := 4;
tj[j25, 9] := 4;   tj[j25,10] := 4;   tj[j25,11] := 4;   tj[j25,12] := 4;
tj[j25,13] := 4;   tj[j25,14] := 4;   tj[j25,15] := 4;   tj[j25,16] := 5;
tj[j25,17] := 5;   tj[j25,18] := 5;   tj[j25,19] := 5;   tj[j25,20] := 5;
tj[j25,21] := 5;   tj[j25,22] := 5;   tj[j25,23] := 5;   tj[j25,24] := 5;
tj[j25,25] := 5;   tj[j25,26] := 5;   tj[j25,27] := 5;   tj[j25,28] := 5;
tj[j25,29] := 5;   tj[j25,30] := 5;   tj[j25,31] := 5;   tj[j25,32] := 5;
tj[j25,33] := 5;   tj[j25,34] := 5;   tj[j25,35] := 4;   tj[j25,36] := 4;

```

end;

OVERLAY procedure temperaturverteilung_3; {kuehles Jahr}

begin

```

tj[ j0, 1] := 2;   tj[ j0, 2] := 0;   tj[ j0, 3] := 0;   tj[ j0, 4] := 0;
tj[ j0, 5] := 0;   tj[ j0, 6] := 0;   tj[ j0, 7] := 0;   tj[ j0, 8] := 2;
tj[ j0, 9] := 4;   tj[ j0,10] := 4;   tj[ j0,11] := 5;   tj[ j0,12] := 6;
tj[ j0,13] := 8;   tj[ j0,14] :=10;   tj[ j0,15] :=12;   tj[ j0,16] :=13;
tj[ j0,17] :=14;   tj[ j0,18] :=15;   tj[ j0,19] :=16;   tj[ j0,20] :=17;
tj[ j0,21] :=18;   tj[ j0,22] :=19;   tj[ j0,23] :=19;   tj[ j0,24] :=18;
tj[ j0,25] :=17;   tj[ j0,26] :=16;   tj[ j0,27] :=15;   tj[ j0,28] :=14;
tj[ j0,29] :=13;   tj[ j0,30] :=12;   tj[ j0,31] :=10;   tj[ j0,32] := 8;
tj[ j0,33] := 6;   tj[ j0,34] := 4;   tj[ j0,35] := 4;   tj[ j0,36] := 3;

```

```

tj[ j5, 1] := 3;   tj[ j5, 2] := 2;   tj[ j5, 3] := 2;   tj[ j5, 4] := 2;
tj[ j5, 5] := 2;   tj[ j5, 6] := 2;   tj[ j5, 7] := 2;   tj[ j5, 8] := 4;
tj[ j5, 9] := 4;   tj[ j5,10] := 4;   tj[ j5,11] := 4;   tj[ j5,12] := 5;
tj[ j5,13] := 7;   tj[ j5,14] := 9;   tj[ j5,15] :=11;   tj[ j5,16] :=12;
tj[ j5,17] :=13;   tj[ j5,18] :=14;   tj[ j5,19] :=15;   tj[ j5,20] :=16;
tj[ j5,21] :=17;   tj[ j5,22] :=18;   tj[ j5,23] :=18;   tj[ j5,24] :=18;
tj[ j5,25] :=17;   tj[ j5,26] :=16;   tj[ j5,27] :=15;   tj[ j5,28] :=14;
tj[ j5,29] :=13;   tj[ j5,30] :=12;   tj[ j5,31] :=10;   tj[ j5,32] := 8;
tj[ j5,33] := 6;   tj[ j5,34] := 4;   tj[ j5,35] := 4;   tj[ j5,36] := 4;

```

```

tj[j10, 1] := 4;   tj[j10, 2] := 3;   tj[j10, 3] := 3;   tj[j10, 4] := 3;
tj[j10, 5] := 3;   tj[j10, 6] := 3;   tj[j10, 7] := 3;   tj[j10, 8] := 3;
tj[j10, 9] := 4;   tj[j10, 10] := 4;   tj[j10, 11] := 4;   tj[j10, 12] := 4;
tj[j10, 13] := 5;   tj[j10, 14] := 7;   tj[j10, 15] := 8;   tj[j10, 16] := 9;
tj[j10, 17] := 10;   tj[j10, 18] := 10;   tj[j10, 19] := 10;   tj[j10, 20] := 11;
tj[j10, 21] := 11;   tj[j10, 22] := 11;   tj[j10, 23] := 12;   tj[j10, 24] := 12;
tj[j10, 25] := 12;   tj[j10, 26] := 12;   tj[j10, 27] := 11;   tj[j10, 28] := 11;
tj[j10, 29] := 11;   tj[j10, 30] := 11;   tj[j10, 31] := 11;   tj[j10, 32] := 10;
tj[j10, 33] := 8;   tj[j10, 34] := 6;   tj[j10, 35] := 4;   tj[j10, 36] := 4;

```

```

tj[j15, 1] := 4;   tj[j15, 2] := 4;   tj[j15, 3] := 3;   tj[j15, 4] := 3;
tj[j15, 5] := 3;   tj[j15, 6] := 3;   tj[j15, 7] := 3;   tj[j15, 8] := 3;
tj[j15, 9] := 4;   tj[j15, 10] := 4;   tj[j15, 11] := 4;   tj[j15, 12] := 4;
tj[j15, 13] := 4;   tj[j15, 14] := 5;   tj[j15, 15] := 6;   tj[j15, 16] := 7;
tj[j15, 17] := 7;   tj[j15, 18] := 8;   tj[j15, 19] := 8;   tj[j15, 20] := 9;
tj[j15, 21] := 9;   tj[j15, 22] := 10;   tj[j15, 23] := 10;   tj[j15, 24] := 10;
tj[j15, 25] := 10;   tj[j15, 26] := 9;   tj[j15, 27] := 8;   tj[j15, 28] := 8;
tj[j15, 29] := 8;   tj[j15, 30] := 8;   tj[j15, 31] := 8;   tj[j15, 32] := 8;
tj[j15, 33] := 6;   tj[j15, 34] := 4;   tj[j15, 35] := 4;   tj[j15, 36] := 4;

```

```

tj[j20, 1] := 4;   tj[j20, 2] := 4;   tj[j20, 3] := 4;   tj[j20, 4] := 4;
tj[j20, 5] := 4;   tj[j20, 6] := 4;   tj[j20, 7] := 4;   tj[j20, 8] := 4;
tj[j20, 9] := 4;   tj[j20, 10] := 4;   tj[j20, 11] := 4;   tj[j20, 12] := 4;
tj[j20, 13] := 4;   tj[j20, 14] := 4;   tj[j20, 15] := 5;   tj[j20, 16] := 5;
tj[j20, 17] := 5;   tj[j20, 18] := 5;   tj[j20, 19] := 5;   tj[j20, 20] := 6;
tj[j20, 21] := 6;   tj[j20, 22] := 6;   tj[j20, 23] := 6;   tj[j20, 24] := 6;
tj[j20, 25] := 6;   tj[j20, 26] := 6;   tj[j20, 27] := 5;   tj[j20, 28] := 5;
tj[j20, 29] := 5;   tj[j20, 30] := 5;   tj[j20, 31] := 5;   tj[j20, 32] := 5;
tj[j20, 33] := 5;   tj[j20, 34] := 4;   tj[j20, 35] := 4;   tj[j20, 36] := 4;

```

```

tj[j25, 1] := 4;   tj[j25, 2] := 4;   tj[j25, 3] := 4;   tj[j25, 4] := 4;
tj[j25, 5] := 4;   tj[j25, 6] := 4;   tj[j25, 7] := 4;   tj[j25, 8] := 4;
tj[j25, 9] := 4;   tj[j25, 10] := 4;   tj[j25, 11] := 4;   tj[j25, 12] := 4;
tj[j25, 13] := 4;   tj[j25, 14] := 4;   tj[j25, 15] := 4;   tj[j25, 16] := 4;
tj[j25, 17] := 4;   tj[j25, 18] := 4;   tj[j25, 19] := 4;   tj[j25, 20] := 4;
tj[j25, 21] := 5;   tj[j25, 22] := 5;   tj[j25, 23] := 5;   tj[j25, 24] := 5;
tj[j25, 25] := 5;   tj[j25, 26] := 5;   tj[j25, 27] := 4;   tj[j25, 28] := 4;
tj[j25, 29] := 4;   tj[j25, 30] := 4;   tj[j25, 31] := 4;   tj[j25, 32] := 4;
tj[j25, 33] := 4;   tj[j25, 34] := 4;   tj[j25, 35] := 4;   tj[j25, 36] := 4;

```

```

end;

```

```

OVERLAY procedure sauerstoffsattigung;
var i: byte; {dekade}
    j: jz;    {Tiefe}
begin
  j := j0;
  while j <= j25 do
    begin
      for i:= 1 to 36 do
        begin
          case tj[j,i] of
            0: osat[j,i] := 14.16;
            1: osat[j,i] := 13.77;
            2: osat[j,i] := 13.40;
            3: osat[j,i] := 13.05;
            4: osat[j,i] := 12.70;
            5: osat[j,i] := 12.37;
            6: osat[j,i] := 12.06;
            7: osat[j,i] := 11.76;
            8: osat[j,i] := 11.47;
            9: osat[j,i] := 11.19;
            10: osat[j,i] := 10.92;
            11: osat[j,i] := 10.67;
            12: osat[j,i] := 10.43;
            13: osat[j,i] := 10.20;
            14: osat[j,i] := 9.98;
            15: osat[j,i] := 9.76;
            16: osat[j,i] := 9.56;
            17: osat[j,i] := 9.37;
            18: osat[j,i] := 9.18;
            19: osat[j,i] := 9.01;
            20: osat[j,i] := 8.84;
            21: osat[j,i] := 8.68;
            22: osat[j,i] := 8.53;
            23: osat[j,i] := 8.38;
            24: osat[j,i] := 8.25;
            25: osat[j,i] := 8.11;
          end;
        end;
      j := succ(j);
    end;
  end;
end;

```

```

===== HINWEISE.OVL =====
Diese Datei enthaelt Prozeduren fuer die waehrend des Programmlaufs auf
dem Bildschirm erscheinenden Hinweise
=====

```

```

OVERLAY procedure interne_duengung (color:boolean);
var co4,col4,col5: byte;

```

```

begin
  writepage(0);
  if color = true
    then begin
      co4 := 4;
      col4 := 14;
      col5 := 15;
    end
    else begin
      co4 := 0;
      col4 := 0;
      col5 := 0;
    end;
  e_gotoxy(16,2,0);textattr(col4,0);write('
  e_gotoxy(16,3,0);          write('
  e_gotoxy(16,4,0);          write('
  e_gotoxy(16,5,0);          write('
  e_gotoxy(16,6,0);          write('
  e_gotoxy(16,7,0);          write('
  e_gotoxy(33,3,0);textattr(col5,0);write('ORTHO-PHOSPHAT');
  e_gotoxy(23,6,0);textattr(co4,0);write('SAUERSTOFFMANGEL');
end;

```

```

FREISETZUNG VON          AUS DEM SEDIMENT
("INTERNE DÜNGUNG")
AM GRUND DES SEES

```

```

OVERLAY procedure hinweis_H2S (color: boolean);
var co4,col4: byte;

```

```

begin
  writepage(0);

```

```

if color = true
  then begin
    co4 := 4;
    col4 := 14;
  end
else begin
  co4 := 0;
  col4 := 0;
end;

e_gotoxy(16,2,0);textattr(col4,0);write('');
e_gotoxy(16,3,0);      write('ENTSTEHUNG GROSSER MENGEN AN');
e_gotoxy(16,4,0);      write('infolge ständiger');
e_gotoxy(16,5,0);      write('Abbauprozesse am Grund des Sees');
e_gotoxy(16,6,0);      write('RAPIDE VERSCHLECHTERUNG DER WASSERQUALITÄT !!');
e_gotoxy(16,7,0);      write('');
e_gotoxy(46,3,0);textattr(co4,0);write('SCHWEFELWASSERSTOFF');
e_gotoxy(36,4,0);textattr(co4,0);write('anaerober (sauerstofffreier)');
end;

```

OVERLAY procedure hinweis_abbruch (color:boolean);

```

var co4,col4: byte;
    i:      integer;

```

```

begin
  writepage(0);
  if color = true
    then begin
      co4 := 4;
      col4 := 14;
    end
  else begin
    co4 := 0;
    col4 := 0;
  end;

  e_gotoxy(16,2,0);textattr(col4,0);write('');
  e_gotoxy(16,3,0);      write('');
  e_gotoxy(16,4,0);      write('');
  e_gotoxy(16,5,0);      write('Giftige Gase und fehlender Sauerstoff führen');
  e_gotoxy(16,6,0);      write('zum Absterben der höheren Organismen !');
  e_gotoxy(16,7,0);      write('');
  e_gotoxy(28,4,0);textattr(co4,0);write('SEE IST "UMGEKIPPT" !!!');

  if color = true then

```

```
for i := 1 to 6 do
  begin
    write(chr(7));
    e_gotoxy(19,3,0); textattr( 0,0); write('ABBRUCH DER SIMULATION!');
    delay(250);
    write(chr(7));
    e_gotoxy(19,3,0); textattr(co4,0); write('ABBRUCH DER SIMULATION!');
    delay(250);
  end;
end;
```

```

===== EGAPROC.INC =====
}

Diese Datei enthaelt die Prozeduren und Funktionen zur Ansteuerung
der EGA-Karte

===== }

{-----}
procedure ega;
EXTERNAL 'EGAPROC';

{----- Grafik-Modus 640 x 350, 4 oder 16 Farben -----}
procedure res_640x350;
type regtyp = record
    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
end;

var reg: regtyp;

begin
    reg.ax := $0010;
    Intr ($10,reg);
end;

{----- Text-Modus 80 x 25, 16 Farben -----}
procedure text_80x25;
type regtyp = record
    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
end;

var reg: regtyp;

begin
    reg.ax := $0003;
    Intr ($10,reg);
end;

{----- Zeichnen eines Punktes -----}
procedure pset (x,y,f,p:Integer);
EXTERNAL ega[0];

{ Parameter:      x:  x-Koordinate des Punktes
                  y:  y-Koordinate des Punktes
                  f:  Farbe des Punktes
                  p:  Bildschirmseite, auf die der Punkt gezeichnet werden
                      soll
}

```

```

{----- Zeichnen einer waagrechten Linie -----}
procedure hline (x1,x2,y,f,p:Integer);
EXTERNAL ega[3];

```

```

{ Parameter:      x1: x-Koordinate, an der die Linie beginnt
                  x2: x-Koordinate, an der die Linie endet
                  y:  y-Koordinate der Linie
                  f:  Farbe des Punktes
                  p:  Bildschirmseite, auf die die Linie gezeichnet werden
                      soll
}

```

```

{----- Zeichnen einer senkrechten Linie -----}
procedure vline (x,y1,y2,f,p:Integer);
EXTERNAL ega[6];

```

```

{ Parameter:      x:  x-Koordinate der Linie
                  y1: y-Koordinate, an der die Linie beginnt
                  y2: y-Koordinate, an der die Linie beginnt
                  f:  Farbe des Punktes
                  p:  Bildschirmseite, auf die die Linie gezeichnet werden
                      soll
}

```

```

{----- Zeichnen einer beliebigen Linie -----}
procedure line (x1,y1,x2,y2,f,p:Integer);
EXTERNAL ega[9];

```

```

{ Parameter:      x1: x-Koordinate, an der die Linie beginnt
                  y1: y-Koordinate, an der die Linie beginnt
                  x2: x-Koordinate, an der die Linie endet
                  y2: y-Koordinate, an der die Linie endet
                  f:  Farbe des Punktes
                  p:  Bildschirmseite, auf die die Linie gezeichnet werden
                      soll
}

```

```

{----- Laden des 8x8 ROM - Zeichensatzes und -----
      Umschalten auf einen Bildschirm mit
      80 Spalten und 43 Zeilen im Textmodus
}

```

```

procedure text_80x43 (cgen:Integer);

```

```

{ Parameter:      cgen: Nummer des Zeichengenerators, in die die
                      Zeichendefinitionen geladen werden soll
}

```

```

type regtyp = record
    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
end;

```

```

var reg: regtyp;

```

```

begin
  text_80x25;
  reg.ax := $1112;
  reg.bx := cgen;
  intr ($10,reg);
end;

{----- Laden des 8x8 ROM-Zeichensatzes; -----}
{ Umschalten auf 80x43 im Grafikmodus }

procedure lines_43;
type regtyp = record
  ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
end;

var reg: regtyp;

begin
  res_640x350;
  reg.ax := $1123;
  reg.bx := $0003;
  intr ($10,reg);
end;

{----- Auswahl der aktiven Seite -----}

procedure selectpage (page:Integer);
{ Parameter      page : Nummer der Seite (1. Seite = 0)  }

type regtyp = record
  ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
end;

var reg: regtyp;

begin
  reg.ax := $0500 OR page;
  Intr ($10,reg);
end;

{----- Bestimmen der aktuellen Cursorposition -----}

procedure e_gotoxy (x,y,p: Integer);
{ Parameter:      x: neue Spaltenposition des Cursors
                  y: neue Zeilenposition des Cursors
                  p: Seite, auf der die Cursorposition verändert
                    werden soll
}

type regtyp = record
  ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
end;

var reg: regtyp;

```

```

begin
  x := x-1;                                { Für Kompatibilität zur Gotoxy-Routine }
  y := y-1;                                { von Turbo-Pascal }
  reg.ax := $0200;
  reg.bx := p shl 8;
  reg.dx := (y shl 8) OR x;
  Intr($10,reg);
end;

{----- Verändern der Farben-Palette -----}
procedure colorpalette (clog,cphys: Integer);

{ Parameter:      clog:    logische Farbe, d.h. die Farbe die bei Zeichen-
                     oder Schreibbefehlen angegeben wird
                  cphys:   physikalische Farbe, d.h. die Farbe die dann
                           tatsächlich auf dem Bildschirm erscheint
}
type regtyp      = record
                    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
                  end;

var reg:         regtyp;

begin
  reg.ax := $1000;
  reg.bx := (cphys shl 8) OR clog;
  Intr ($10,reg);
end;

{----- Lesen des Umfangs des EGA-Speichers -----}
function egamemory:Integer;
{ Parameter: Die Funktion übergibt den Umfang des EGA-Speichers in KByte}
type regtyp      = record
                    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
                  end;

var reg:         regtyp;

begin
  reg.ax := $1200;
  reg.bx := $0010;
  Intr ($10,reg);
  egamemory := (reg.bx AND 255)*64 + 64;
end;

```

```

{----- Lesen der Farbe eines Bildschirmpunktes -----}
function pointcolor (x,y,p: Integer): Integer;
{
  Parameter:      x:      x-Koordinate des Bildschirmpunktes
                  y:      y-Koordinate des Bildschirmpunktes
                  p:      Seite auf der zu lesende Bildschirmpunkt sich
                          befindet
}
type regtyp      = record
                    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
                    end;
var reg:          regtyp;

begin
  reg.ax := $0D00;
  reg.bx := p SHL 8;
  reg.cx := x;
  reg.dx := y;
  Intr ($10,reg);
  pointcolor := reg.ax AND 255;
end;

{--- Lesen der aktuellen Position des Cursors auf einer beliebigen Seite ---}
procedure getcursorpos (var x,y:Integer; p:Integer);
{
  Parameter:      x:      aktuelle Spalte des Cursors (Ausgabeparameter)
                  y:      aktuelle Zeile des Cursors (Ausgabeparameter)
                  p:      Seite, von der die Cursorposition gelesen
                          werden soll
}
type regtyp      = record
                    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
                    end;
var reg:          regtyp;

begin
  reg.ax := $0300;
  reg.bx := p SHL 8;
  Intr ($10,reg);
  x := lo(reg.dx)+1;           { "+1" für Kompatibilität zu Wherex }
  y := hi(reg.dx)+1;         { "+1" für Kompatibilität zu Wherey }
end;

{----- Lesen der Nummer der aktiven Seite -----}
function activepage: Integer;
type regtyp      = record
                    ax,bx,cx,dx,bp,si,di,ds,es,flags: Integer;
                    end;
var reg:          regtyp;

```

```

begin
  reg.ax := $0F00;
  Intr ($10,reg);
  activepage := (reg.bx SHR 8) AND 255;
end;

{----- BildschirmAusgabe-Routine -----}
procedure text_write (c:Char);

type regtyp = record
  ax,bx,cx,dx, bp,si,di,ds,es,flags: Integer;
end;

var reg:      regtyp;
    ap,
    xpos,
    ypos:     Integer;
    z:        Char;
    spezial:   Boolean;

begin
  z := c;
  spezial := Ord(z) IN [8,10,13];
  if spezial then
    begin
      ap := activepage;
      selectpage (e_text_page);
    end;
  reg.ax := $0301;
  reg.bx := e_text_page SHL 8;
  Intr ($10,reg);
  reg.ax := $1301;
  reg.bx := (e_text_page SHL 8) OR e_text_attribute;
  reg.cx := 1;
  reg.bp := Ofz(z);
  reg.es := Seg(z);
  Intr ($10,reg);
  if spezial then
    selectpage (ap);
end;

{----- Ersetzen der TURBO-PASCAL-BildschirmAusgaberroutine -----}
      durch die EGAPROC.OVR-Routine text-write
}
procedure text_bypass;
begin
  conoutptr := ofs(text_write);
  e_text_page := 0;
  e_text_attribute := 14;
end;

```

```
{----- Wahl der Farbe eines Zeichens -----}
procedure textattr (ccolor,bcolor:Integer);
```

```
{Parameter:      ccolor:   Farbe des Zeichens (0-15)
                  bcolor:   Farbe des Hintergrunds des Zeichens (0-15)
}
```

```
begin
e_text_attribute := (bcolor SHL 8) OR ccolor;
end;
```

```
{----- Auswahl der Seite, auf die geschrieben werden soll -----}
procedure writepage (page:Integer);
```

```
begin
  e_text_page := page;
end;
```

```
{----- Warten auf Eingabe von Konsole -----}
procedure e_press_key (c,r,p:integer);
```

```
var  ap,x,y: integer;
      z:      char;
```

```
begin
  getcursorpos (x,y,p);
  ap := activepage;
  e_gotoxy (c,r,p); textattr(14,0);
  writepage (p);
  Writeln ('Weiter: Bitte eine Taste drücken');
  repeat until keypressed;
  read (Kbd,z);
  writepage (ap);
  e_gotoxy (x,y,ap);
end;
```

 FONT4x6.INC

Diese Datei enthaelt Prozeduren zur Erzeugung der aus 4x6 Pixel
zusammengesetzten Schrift

```

procedure AA (spx,spy,c,p:integer);
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;

procedure AE (spx,spy,c,p:integer);
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+3,spy+4,c,p);
  pset(spx ,spy-1,c,p); pset(spx+3,spy-1,c,p);
end;

procedure BB (spx,spy,c,p:integer);
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx ,spy+1,c,p); pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+1,spy+4,c,p);
  pset(spx+2,spy+4,c,p);
end;

procedure CC (spx,spy,c,p:integer);
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx+3,spy ,c,p);
  pset(spx ,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;

procedure DD (spx,spy,c,p:integer);
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx+1,spy+1,c,p); pset(spx+3,spy+1,c,p); pset(spx+1,spy+2,c,p);
  pset(spx+3,spy+2,c,p); pset(spx+1,spy+3,c,p); pset(spx+3,spy+3,c,p);
  pset(spx ,spy+4,c,p); pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p);
end;

```

```
procedure EE (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+1,spy+4,c,p);
  pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;
```

```
procedure FF (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx ,spy+4,c,p);
end;
```

```
procedure GG (spx,spy,c,p:integer);
```

```
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx+3,spy ,c,p);
  pset(spx ,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p);
end;
```

```
procedure HH (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;
```

```
procedure II (spx,spy,c,p:integer);
```

```
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx+3,spy ,c,p);
  pset(spx+2,spy+1,c,p); pset(spx+2,spy+2,c,p); pset(spx+2,spy+3,c,p);
  pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;
```

```
procedure JJ (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx+3,spy ,c,p); pset(spx+3,spy+1,c,p); pset(spx+3,spy+2,c,p);
  pset(spx+3,spy+3,c,p); pset(spx+2,spy+4,c,p); pset(spx+1,spy+4,c,p);
  pset(spx ,spy+3,c,p);
end;
```

```
procedure KK (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+2,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx+2,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+3,spy+4,c,p);
end;
```

```
procedure LL (spx,spy,c,p:integer);
```

```
begin
```

```
  pset(spx ,spy ,c,p); pset(spx ,spy+1,c,p); pset(spx ,spy+2,c,p);
```

```
  pset(spx ,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+1,spy+4,c,p);
```

```
  pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
```

```
end;
```

```
procedure MM (spx,spy,c,p:integer); {5}
```

```
begin
```

```
  pset(spx ,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p);
```

```
  pset(spx+1,spy+1,c,p); pset(spx+2,spy+1,c,p); pset(spx+3,spy+1,c,p);
```

```
  pset(spx ,spy+2,c,p); pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p);
```

```
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+3,spy+4,c,p);
```

```
end;
```

```
procedure NN (spx,spy,c,p:integer);
```

```
begin
```

```
  pset(spx ,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p);
```

```
  pset(spx+1,spy+1,c,p); pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p);
```

```
  pset(spx+2,spy+2,c,p); pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p);
```

```
  pset(spx+2,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p);
```

```
  pset(spx+3,spy+4,c,p);
```

```
end;
```

```
procedure OO (spx,spy,c,p:integer);
```

```
begin
```

```
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx ,spy+1,c,p);
```

```
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+3,spy+2,c,p);
```

```
  pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx+1,spy+4,c,p);
```

```
  pset(spx+2,spy+4,c,p);
```

```
end;
```

```
procedure OE (spx,spy,c,p:integer);
```

```
begin
```

```
  pset(spx ,spy-1,c,p); pset(spx+3,spy-1,c,p);
```

```
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx ,spy+1,c,p);
```

```
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+3,spy+2,c,p);
```

```
  pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx+1,spy+4,c,p);
```

```
  pset(spx+2,spy+4,c,p);
```

```
end;
```

```
procedure PP (spx,spy,c,p:integer);
```

```
begin
```

```
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
```

```
  pset(spx ,spy+1,c,p); pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p);
```

```
  pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p); pset(spx ,spy+3,c,p);
```

```
  pset(spx ,spy+4,c,p);
```

```
end;
```

```
procedure RR (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx ,spy+1,c,p); pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;
```

```
procedure SS (spx,spy,c,p:integer);
```

```
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx+3,spy ,c,p);
  pset(spx ,spy+1,c,p); pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p);
  pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p); pset(spx+1,spy+4,c,p);
  pset(spx+2,spy+4,c,p);
end;
```

```
procedure TT (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p);
  pset(spx+3,spy ,c,p); pset(spx+4,spy ,c,p); pset(spx+2,spy+1,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+2,spy+3,c,p); pset(spx+1,spy+4,c,p);
  pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;
```

```
procedure UU (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+3,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
end;
```

```
procedure UE (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+3,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p); pset(spx+3,spy+4,c,p);
  pset(spx+1,spy-1,c,p); pset(spx+2,spy-1,c,p);
end;
```

```
procedure WW (spx,spy,c,p:integer);{5}
```

```
begin
  pset(spx ,spy ,c,p); pset(spx+4,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+4,spy+1,c,p); pset(spx ,spy+2,c,p); pset(spx+2,spy+2,c,p);
  pset(spx+4,spy+2,c,p); pset(spx ,spy+3,c,p); pset(spx+1,spy+3,c,p);
  pset(spx+3,spy+3,c,p); pset(spx+4,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+4,spy+4,c,p);
end;
```

```
procedure XX (spx,spy,c,p:integer); {5}
```

```
begin
  pset (spx ,spy ,c,p); pset (spx+4,spy ,c,p); pset (spx+1,spy+1,c,p);
  pset (spx+3,spy+1,c,p); pset (spx+2,spy+2,c,p); pset (spx+1,spy+3,c,p);
  pset (spx+3,spy+3,c,p); pset (spx ,spy+4,c,p); pset (spx+4,spy+4,c,p);
end;
```

```
procedure ZZ (spx,spy,c,p:integer); {5}
```

```
begin
  pset (spx ,spy ,c,p); pset (spx+1,spy ,c,p); pset (spx+2,spy ,c,p);
  pset (spx+3,spy ,c,p); pset (spx+4,spy ,c,p); pset (spx+3,spy+1,c,p);
  pset (spx+2,spy+2,c,p); pset (spx+1,spy+3,c,p); pset (spx ,spy+4,c,p);
  pset (spx+1,spy+4,c,p); pset (spx+2,spy+4,c,p); pset (spx+3,spy+4,c,p);
  pset (spx+4,spy+4,c,p);
end;
```

```
procedure klein_c (spx,spy,c,p:integer);
```

```
begin
  pset (spx+1,spy+1,c,p); pset (spx+2,spy+1,c,p); pset (spx+3,spy+1,c,p);
  pset (spx ,spy+2,c,p); pset (spx ,spy+3,c,p); pset (spx+1,spy+4,c,p);
  pset (spx+2,spy+4,c,p); pset (spx+3,spy+4,c,p);
end;
```

```
procedure klein_d (spx,spy,c,p:integer);
```

```
begin
  pset (spx+3,spy ,c,p); pset (spx+1,spy+1,c,p); pset (spx+2,spy+1,c,p);
  pset (spx+3,spy+1,c,p); pset (spx ,spy+2,c,p); pset (spx+3,spy+2,c,p);
  pset (spx ,spy+3,c,p); pset (spx+3,spy+3,c,p); pset (spx+1,spy+4,c,p);
  pset (spx+2,spy+4,c,p); pset (spx+3,spy+4,c,p);
end;
```

```
procedure klein_g (spx,spy,c,p:integer);
```

```
begin
  pset (spx+1,spy+1,c,p); pset (spx+2,spy+1,c,p); pset (spx+3,spy+1,c,p);
  pset (spx ,spy+2,c,p); pset (spx+3,spy+2,c,p); pset (spx+1,spy+3,c,p);
  pset (spx+2,spy+3,c,p); pset (spx+3,spy+3,c,p); pset (spx+3,spy+4,c,p);
  pset (spx ,spy+5,c,p); pset (spx+1,spy+5,c,p); pset (spx+2,spy+5,c,p);
  pset (spx+3,spy+5,c,p);
end;
```

```
procedure klein_k (spx,spy,c,p:integer);
```

```
begin
  pset (spx ,spy ,c,p); pset (spx ,spy+1,c,p); pset (spx ,spy+2,c,p);
  pset (spx+2,spy+2,c,p); pset (spx ,spy+3,c,p); pset (spx+1,spy+3,c,p);
  pset (spx ,spy+4,c,p); pset (spx+2,spy+4,c,p);
end;
```

```
procedure klein_l (spx,spy,c,p:integer);
```

```
begin
  pset (spx+1,spy ,c,p); pset (spx+1,spy+1,c,p); pset (spx+1,spy+2,c,p);
  pset (spx+1,spy+3,c,p); pset (spx ,spy+4,c,p); pset (spx+1,spy+4,c,p);
  pset (spx+2,spy+4,c,p);
end;
```

```
procedure klein_m (spx,spy,c,p:integer); {5}
```

```
begin
  pset(spx ,spy+1,c,p); pset(spx+1,spy+1,c,p); pset(spx+2,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx+4,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+4,spy+2,c,p); pset(spx ,spy+3,c,p);
  pset(spx+2,spy+3,c,p); pset(spx+4,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+2,spy+4,c,p); pset(spx+4,spy+4,c,p);
end;
```

```
procedure klein_n (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy+1,c,p); pset(spx+2,spy+1,c,p); pset(spx+3,spy+1,c,p);
  pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p); pset(spx+3,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+3,spy+4,c,p);
end;
```

```
procedure klein_u (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy+1,c,p); pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx+3,spy+2,c,p); pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p);
  pset(spx ,spy+4,c,p); pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p);
  pset(spx+3,spy+4,c,p);
end;
```

```
procedure pkt (spx,spy,c,p:integer);
```

```
begin
  pset(spx+2,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx+2,spy+4,c,p);
  pset(spx+3,spy+4,c,p);
end;
```

```
procedure trstr (spx,spy,c,p:integer);
```

```
begin
  pset(spx ,spy+2,c,p); pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p);
end;
```

```
procedure sstr (spx,spy,c,p:integer);
```

```
begin
  pset(spx+4,spy ,c,p); pset(spx+3,spy+1,c,p); pset(spx+2,spy+2,c,p);
  pset(spx+1,spy+3,c,p); pset(spx ,spy+4,c,p);
end;
```

```
procedure grad (spx,spy,c,p:integer);
```

```
begin
  pset(spx+2,spy ,c,p); pset(spx+3,spy ,c,p); pset(spx+4,spy ,c,p);
  pset(spx+5,spy ,c,p); pset(spx+1,spy+1,c,p); pset(spx+2,spy+1,c,p);
  pset(spx+5,spy+1,c,p); pset(spx+6,spy+1,c,p); pset(spx+1,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+5,spy+2,c,p); pset(spx+6,spy+2,c,p);
  pset(spx+2,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx+4,spy+3,c,p);
  pset(spx+5,spy+3,c,p);
end;
```

```

procedure mal (spx,spy,c,p:integer);
begin
  pset (spx+1,spy+2,c,p); pset (spx+2,spy+2,c,p); pset (spx+1,spy+3,c,p);
  pset (spx+2,spy+3,c,p);
end;

procedure z1 (spx,spy,c,p:integer);
begin
  pset (spx+1,spy ,c,p); pset (spx ,spy+1,c,p); pset (spx+1,spy+1,c,p);
  pset (spx+1,spy+2,c,p); pset (spx+1,spy+3,c,p); pset (spx ,spy+4,c,p);
  pset (spx+1,spy+4,c,p); pset (spx+2,spy+4,c,p);
end;

procedure z2 (spx,spy,c,p:integer);
begin
  pset (spx ,spy ,c,p); pset (spx+1,spy ,c,p); pset (spx+2,spy+1,c,p);
  pset (spx+1,spy+2,c,p); pset (spx ,spy+3,c,p); pset (spx ,spy+4,c,p);
  pset (spx+1,spy+4,c,p); pset (spx+2,spy+4,c,p);
end;

procedure z3 (spx,spy,c,p:integer);
begin
  pset (spx+1,spy ,c,p); pset (spx+2,spy ,c,p); pset (spx ,spy+1,c,p);
  pset (spx+3,spy+1,c,p); pset (spx+2,spy+2,c,p); pset (spx ,spy+3,c,p);
  pset (spx+3,spy+3,c,p); pset (spx+1,spy+4,c,p); pset (spx+2,spy+4,c,p);
end;

procedure z4 (spx,spy,c,p:integer);
begin
  pset (spx ,spy ,c,p); pset (spx ,spy+1,c,p); pset (spx+2,spy+1,c,p);
  pset (spx ,spy+2,c,p); pset (spx+1,spy+2,c,p); pset (spx+2,spy+2,c,p);
  pset (spx+3,spy+2,c,p); pset (spx+2,spy+3,c,p); pset (spx+2,spy+4,c,p);
end;

procedure z5 (spx,spy,c,p:integer);
begin
  pset (spx ,spy ,c,p); pset (spx+1,spy ,c,p); pset (spx+2,spy ,c,p);
  pset (spx ,spy+1,c,p); pset (spx ,spy+2,c,p); pset (spx+1,spy+2,c,p);
  pset (spx+2,spy+3,c,p); pset (spx ,spy+4,c,p); pset (spx+1,spy+4,c,p);
end;

procedure z6 (spx,spy,c,p:integer);
begin
  pset (spx+2,spy ,c,p); pset (spx+1,spy+1,c,p); pset (spx ,spy+2,c,p);
  pset (spx+2,spy+2,c,p); pset (spx ,spy+3,c,p); pset (spx+1,spy+3,c,p);
  pset (spx+1,spy+4,c,p); pset (spx+2,spy+4,c,p);
end;

procedure z7 (spx,spy,c,p:integer);
begin
  pset (spx ,spy ,c,p); pset (spx+1,spy ,c,p); pset (spx+2,spy ,c,p);
  pset (spx+3,spy ,c,p); pset (spx ,spy+1,c,p); pset (spx+3,spy+1,c,p);
  pset (spx+2,spy+2,c,p); pset (spx+1,spy+3,c,p); pset (spx+1,spy+4,c,p);
end;

```

```

procedure z8 (spx,spy,c,p:integer);
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p);
  pset(spx ,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx+1,spy+4,c,p);
  pset(spx+2,spy+4,c,p);
end;

```

```

procedure z9 (spx,spy,c,p:integer);
begin
  pset(spx+1,spy ,c,p); pset(spx+2,spy ,c,p); pset(spx ,spy+1,c,p);
  pset(spx+3,spy+1,c,p); pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p);
  pset(spx+3,spy+2,c,p); pset(spx+3,spy+3,c,p); pset(spx ,spy+4,c,p);
  pset(spx+1,spy+4,c,p); pset(spx+2,spy+4,c,p);
end;

```

```

procedure pfeilr (spx,spy,c,p:integer); {5}
begin
  pset(spx+2,spy ,c,p); pset(spx+3,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx+1,spy+2,c,p); pset(spx+2,spy+2,c,p); pset(spx+3,spy+2,c,p);
  pset(spx+4,spy+2,c,p); pset(spx+3,spy+3,c,p); pset(spx+2,spy+4,c,p);
end;

```

```

procedure pfeilu (spx,spy,c,p:integer); {5}
begin
  pset(spx+2,spy ,c,p); pset(spx+2,spy+1,c,p); pset(spx ,spy+2,c,p);
  pset(spx+2,spy+2,c,p); pset(spx+4,spy+2,c,p); pset(spx+1,spy+3,c,p);
  pset(spx+2,spy+3,c,p); pset(spx+3,spy+3,c,p); pset(spx+2,spy+4,c,p);
end;

```

```

{ ===== TSCHICHT.INC ===== }

      Diese Datei enthält die Prozeduren
      zum Auf- und Abbau der thermischen Schichtung

}

{ ===== Zeichne Linie innerhalb des Seebeckens ===== }
procedure hatchline(line,f,p:byte);
begin
  hline((win[3,xol]+hxl[line]),(win[3,xol]+hxr[line]),(win[3,yol]+line),f,p);
end;

{ ===== Schraffiere aufwärts ===== }
procedure hatchrowup(rowanf,rowend,co,pa:byte);
var row: byte;
begin
  for row := rowanf to rowend do hatchline(row,co,pa);
end;

{ ===== Schraffiere abwärts ===== }
procedure hatchrowdown(rowanf,rowend,co,pa:byte);
var row: byte;
begin
  for row := rowanf downto rowend do hatchline(row,co,pa);
end;

{ ===== Zeichne Durchmischungspfeile ===== }
procedure flash_vollzirkulation(co: byte);
const xend2 = 272; {Endkoordinaten der Kreisteile 2,3,5,6}
      yend2 = 154;
      xend3 = 205;
      yend3 = 121;
      xend5 = 354;
      yend5 = 78;
      xend6 = 427;
      yend6 = 79;

var i: integer;

begin
  {Pfeile aus array zeichnen}
  for i := 255 to 390 do pset(flashx1[i],flashy1[i],co,0);
  for i := 450 to 555 do pset(flashx2[i],flashy2[i],co,0);
  for i := 315 to 495 do pset(flashx3[i],flashy3[i],co,0);
  for i := 699 to 804 do pset(flashx4[i],flashy4[i],co,0);
  for i := 864 to 999 do pset(flashx5[i],flashy5[i],co,0);
  for i := 756 to 936 do pset(flashx6[i],flashy6[i],co,0);
  {Text O2 und PO4}
  OO(win[3,xol]+109 ,win[3,yol]+49,co,0);
  z2(win[3,xol]+109+ 6,win[3,yol]+52,co,0);
  PP(win[3,xol]+284 ,win[3,yol]+51,co,0);
  OO(win[3,xol]+284+ 6,win[3,yol]+51,co,0);
  z4(win[3,xol]+284+12,win[3,yol]+54,co,0);

```

```

{Pfeilspitzen}
{Kreisteil 2}
hline(xend2,xend2-12,yend2,co,0);
line(xend2,yend2,xend2-6,yend2-6,co,0);
line(xend2+1,yend2,xend2+1-6,yend2-6,co,0);
{Kreisteil 3}
hline(xend3,xend3-7,yend3,co,0);
vline(xend3,yend3,yend3-5,co,0);
vline(xend3+1,yend3,yend3-5,co,0);
{Kreisteil 5}
line(xend5,yend5,xend5+12,yend5-3,co,0);
line(xend5,yend5,xend5+ 7,yend5+7,co,0);
line(xend5-1,yend5,xend5-1+ 7,yend5+7,co,0);
{Kreisteil 6}
line(xend6,yend6,xend6+9,yend6-2,co,0);
vline(xend6,yend6,yend6+6,co,0);
end;

{ ----- thermische Schichtung bei warmem Sommer ----- }
procedure schraffierung1;
const coepi   = 11;
      comet   = 9;
      cohyp   = 1;
      coflash = 1;

begin
  case deka of
    1: begin
        if jahr > 1 then colorpalette(11,0); {Schraffierung nicht sichtbar}
        hatchrowdown(100, 4,coepi,0);      {Vollzirkulation}
        if jahr > 1 then colorpalette (11,11); {Schraff. sichtbar machen}
      end;
    2: hatchrowup ( 4, 6,cohyp,0); {Beginn der Winterschichtung}
    3: hatchrowup ( 7, 8,cohyp,0); {+ Winterschichtung}
    4: hatchrowup ( 9,10,cohyp,0); {max Winterschichtung}
    6: hatchrowdown( 10, 9,coepi,0); {- Winterschichtung}
    7: hatchrowdown( 8, 6,coepi,0); {- Winterschichtung}
    8: begin
        hatchrowdown( 5, 4,coepi,0); {Vollzirkulation}
        flash_vollzirkulation(coflash);
      end;
    10: begin
        flash_vollzirkulation(coepi); {loeschen}
        hatchrowdown(100,92,cohyp,0); {Stag1 +H25-23}
      end;
    11: hatchrowdown( 91,84,cohyp,0); {Stag2 +H23-21}
    12: hatchrowdown( 83,73,cohyp,0); {Stag3 +H21-18}
    13: hatchrowdown( 72,60,cohyp,0); {Stag4 +H18-15}
    14: hatchrowdown( 59,48,comet,0); {Stag4 +H15-12}
    15: begin
        hatchrowdown(47,40,comet,0);
        hatchrowdown(59,52,cohyp,0);
      end;
  end;

```

```

17: begin                                     {Stag6 +H10- 7, +H13-12}
    hatchrowdown(39,28,comet,0);
    hatchrowdown(51,48,cohyp,0);
end;
29: hatchrowup ( 28,39,coepi,0);           {Stag7 +E 7-10}
31: hatchrowup ( 40,59,coepi,0);           {Stag8 +E10-15}
33: hatchrowup ( 60,79,coepi,0);           {Stag9 +E15-20}
34: begin
    hatchrowup ( 80,100,coepi,0); {Vollzirkulation}
    flash_vollzirkulation(coflash);
end;
36: flash_vollzirkulation(coepi);
end;
end;

{ ----- thermische Schichtung bei normalem Sommer ----- }
procedure schraffierung2;
const coepi = 11;
      comet = 9;
      cohyp = 1;
      coflash = 1;

begin
  case deka of
    1: begin
        if jahr > 1 then colorpalette(11,0); {Schraffierung nicht sichtbar}
        hatchrowdown(100, 4,coepi,0);        {Vollzirkulation}
        if jahr > 1 then colorpalette (11,11); {Schraff. sichtbar machen}
      end;
    2: hatchrowup ( 4, 6,cohyp,0);           {Beginn der Winterschichtung}
    3: hatchrowup ( 7, 9,cohyp,0);           {+ Winterschichtung}
    4: hatchrowup ( 9,11,cohyp,0);           {max Winterschichtung}
    6: hatchrowdown( 11,10,coepi,0);         {- Winterschichtung}
    7: hatchrowdown( 9, 7,coepi,0);          {- Winterschichtung}
    8: begin
        hatchrowdown(6, 4,coepi,0);          {Vollzirkulation}
        flash_vollzirkulation(coflash);
      end;
    10: begin
        flash_vollzirkulation(coepi); {loeschen}
        hatchrowdown(100,92,cohyp,0); {Stag1 +H25-23}
      end;
    11: hatchrowdown( 91,84,cohyp,0);         {Stag2 +H23-21}
    12: hatchrowdown( 83,73,cohyp,0);         {Stag2 +H21-18}
    13: hatchrowdown( 72,60,cohyp,0);         {Stag3 +H18-15}
    14: hatchrowdown( 59,48,comet,0);         {Stag4 +H15-12}
    17: begin
        hatchrowdown(47,40,comet,0);
        hatchrowdown(59,52,cohyp,0);
      end;
    28: hatchrowup ( 40,47,coepi,0);         {Stag6 +E10-12}
    31: hatchrowup ( 48,59,coepi,0);         {Stag7 +E12-15}
    33: hatchrowup ( 60,79,coepi,0);         {Stag8 +E15-20}
  end;
end;

```

```

34: begin
    hatchrowup ( 80,100,coepi,0); {Vollzirkulation}
    flash_vollzirkulation(coflash);
end;
36: flash_vollzirkulation(coepi);
end;
end;

{ ----- thermische Schichtung bei kuehlem Sommer ----- }
procedure schraffierung3;
const coepi = 11;
      comet = 9;
      cohyp = 1;
      coflash = 1;

begin
  case deka of
    1: begin
        if jahr > 1 then colorpalette(11,0); {Schraffierung nicht sichtbar}
        hatchrowdown(100, 4,coepi,0); {Vollzirkulation}
        if jahr > 1 then colorpalette (11,11); {Schraff. sichtbar machen}
        end;
    2: hatchrowup ( 4, 6,cohyp,0); {Beginn der Winterschichtung}
    3: hatchrowup ( 7, 9,cohyp,0); {+ Winterschichtung}
    4: hatchrowup ( 10,11,cohyp,0); {+ Winterschichtung}
    5: hatchrowup ( 12,13,cohyp,0); {max Winterschichtung}
    7: hatchrowdown( 13,11,coepi,0); {- Winterschichtung}
    8: hatchrowdown( 10, 8,coepi,0); {- Winterschichtung}
    9: begin
        hatchrowdown( 7, 4,coepi,0); {Vollzirkulation}
        flash_vollzirkulation(coepi);
        end;
    13: begin
        flash_vollzirkulation(coepi); {loeschen}
        hatchrowdown(100,92,cohyp,0); {Stag1 +H25-23}
        end;
    14: hatchrowdown( 91,83,cohyp,0); {Stag2 +H23-21}
    15: hatchrowdown( 82,72,cohyp,0); {Stag2 +H21-18}
    17: hatchrowdown( 71,60,cohyp,0); {Stag3 +H18-15}
    19: hatchrowdown( 59,52,comet,0); {Stag4 +H15-13}
    29: hatchrowup ( 52,59,coepi,0); {Stag5 +E13-15}
    31: hatchrowup ( 60,71,coepi,0); {Stag6 +E15-18}
    32: hatchrowup ( 72,79,coepi,0); {Stag7 +E18-20}
    34: begin
        hatchrowup ( 80,100,coepi,0); {Vollzirkulation}
        flash_vollzirkulation(coflash);
        end;
    36: flash_vollzirkulation(coepi);
  end;
end;
end;

```

LAUFZEITANALYSE
DES COMPUTERPROGRAMMS

total time: 826.6197 seconds total calls: 10420

% time	# calls	recurs	total time	time/call	procedure/function
1.5	1	1	12.2032	12.2032	lago_EGA
0.2	4	1	1.7430	0.4357	res_640x350
0.2	6	1	1.6359	0.2726	text_80x25
0.0	0	0	0.0000		text_80x43
0.0	2	1	0.0034	0.0017	lines_43
0.0	638	1	0.0221	0.0000	selectpage
0.1	706	1	1.0818	0.0015	e_gotoxy
0.0	8	1	0.0131	0.0016	colorpalette
0.0	0	0	0.0000		egamemory
0.0	1	1	0.0019	0.0019	pointcolor
0.0	1	1	0.0013	0.0013	getcursorpos
0.0	319	1	0.0077	0.0000	activepage
4.3	15409	1	35.2641	0.0023	text_write
0.0	8	1	0.0026	0.0003	text_bypass
0.0	223	1	0.0640	0.0003	textattr
0.0	124	1	0.0330	0.0003	writepage
0.2	1	1	1.3101	1.3101	e_press_key
0.0	13	1	0.1886	0.0145	AA
0.0	0	0	0.0000		AE
0.0	2	1	0.0311	0.0156	BB
0.0	14	1	0.1545	0.0110	CC
0.0	11	1	0.1599	0.0145	DD
0.0	19	1	0.3213	0.0169	EE
0.0	9	1	0.1210	0.0134	FF
0.0	0	0	0.0000		GG
0.0	21	1	0.3034	0.0144	HH
0.0	15	1	0.1642	0.0109	II
0.0	15	1	0.1824	0.0122	JJ
0.0	1	1	0.0120	0.0120	KK
0.0	2	1	0.0195	0.0098	LL
0.0	13	1	0.1886	0.0145	MM
0.0	16	1	0.2511	0.0157	NN
0.1	80	1	0.9683	0.0121	OO
0.0	0	0	0.0000		OE
0.1	48	1	0.5831	0.0121	PP
0.0	14	1	0.2024	0.0145	RR
0.0	22	1	0.2665	0.0121	SS
0.0	16	1	0.2133	0.0133	TT
0.0	8	1	0.1159	0.0145	UU
0.0	1	1	0.0172	0.0172	UE
0.0	2	1	0.0318	0.0159	WW
0.0	0	0	0.0000		XX
0.0	2	1	0.0311	0.0156	ZZ
0.0	7	1	0.0680	0.0097	klein_c
0.0	5	1	0.0674	0.0135	klein_d
0.0	5	1	0.0792	0.0158	klein_g
0.0	1	1	0.0099	0.0099	klein_k
0.0	1	1	0.0086	0.0086	klein_l
0.1	26	1	0.4389	0.0169	klein_m
0.0	2	1	0.0245	0.0122	klein_n
0.0	2	1	0.0240	0.0120	klein_u
0.0	10	1	0.0502	0.0050	pkt
0.0	3	1	0.0112	0.0037	trstr
0.0	12	1	0.0738	0.0062	sstr
0.0	0	0	0.0000		grad
0.0	3	1	0.0148	0.0049	mal
0.0	12	1	0.1180	0.0098	z1

262	0.0	17	1	0.1646	0.0097	z2
	0.0	17	1	0.1869	0.0110	z3
	0.1	46	1	0.5060	0.0110	z4
	0.0	8	1	0.0871	0.0109	z5
	0.0	4	1	0.0397	0.0099	z6
	0.0	1	1	0.0109	0.0109	z7
	0.0	2	1	0.0242	0.0121	z8
	0.0	2	1	0.0268	0.0134	z9
	0.0	1	1	0.0109	0.0109	pfeilr
	0.0	1	1	0.0112	0.0112	pfeilu
	7.6	269	1	62.9759	0.2341	hatchline
	0.0	7	1	0.0449	0.0064	hatchrowup
	0.0	13	1	0.1000	0.0077	hatchrowdown
	0.7	4	1	6.0573	1.5143	flash_vollzirku
	0.0	36	1	0.0307	0.0009	schraffierung1
	0.0	0	0	0.0000		schraffierung2
	0.0	0	0	0.0000		schraffierung3
	0.0	1	1	0.0034	0.0034	temperaturverte
	0.0	0	0	0.0000		temperaturverte
	0.0	0	0	0.0000		temperaturverte
	0.0	1	1	0.1315	0.1315	sauerstoffsaeft
	0.0	12	1	0.0300	0.0025	flush_keyboard
	0.0	145	1	0.0779	0.0005	pruefstop
	3.0	6	1	24.9932	4.1655	blink_cursor
	0.0	1	1	0.4077	0.4077	logo
	0.5	1	1	4.5163	4.5163	read_datei_wate
	0.1	1	1	0.7689	0.7689	eingabe_trophie
	0.1	1	1	1.0037	1.0037	eingabe_klima
	0.1	1	1	1.0488	1.0488	phosphatbilanz
	0.1	1	1	0.5509	0.5509	text_seedorf
	0.0	1	1	0.0702	0.0702	berechnung_und_
	0.0	0	0	0.0000		eingabefeld
	0.0	0	0	0.0000		aendern
	0.0	0	0	0.0000		einwohner_neu
	0.0	0	0	0.0000		pzsatz_neu
	0.0	0	0	0.0000		landwflaeche_ne
	0.0	0	0	0.0000		entsorgung_neu
	0.0	0	0	0.0000		berechnungsgrun
	0.0	0	1	0.0000		menue_und_verzw
	0.0	0	0	0.0000		bypass_ovl_menu
	0.0	1	1	0.1363	0.1363	morphometrie
	0.0	5	1	0.2039	0.0408	vole_j
	2.6	1	1	21.2802	21.2802	draw_grafik
	0.2	3	1	1.6050	0.5350	text_kJ
	0.1	2	1	0.8387	0.4193	text_mg_m3
	0.2	4	1	1.5868	0.3967	text_cm_m3
	0.1	1	1	0.4133	0.4133	text_ortho
	0.4	2	1	3.2602	1.6301	phosphat_XXport
	0.1	2	1	0.9273	0.4636	text_ortho
	0.1	2	1	0.8700	0.4350	text_mg_m3
	0.1	1	1	0.6771	0.6771	abbruch_bestaat
	0.1	1	1	0.8281	0.8281	menu_weiter
	0.0	0	0	0.0000		experimente
	0.0	0	0	0.0000		text_experiment
	0.0	0	0	0.0000		eingabefeld2
	0.0	0	0	0.0000		aendern
	0.0	0	0	0.0000		tempneu
	0.0	0	0	0.0000		tempzneu
	0.0	0	0	0.0000		pxkoeffneu
	0.0	0	0	0.0000		pzkoeffneu
	0.0	0	0	0.0000		faktphys_neu
	0.0	0	0	0.0000		lichthemm_neu
	0.0	0	0	0.0000		vxsed_neu

0.0	0	0	0.0000		verzweigung
0.0	0	0	0.0000		bypass_verzweig
0.0	0	1	0.0000		menu_laughtyp
0.0	2	1	0.3311	0.1656	interne_duengun
0.0	0	0	0.0000		hinweis_H2S
0.0	0	0	0.0000		hinweis_abbruch
0.0	145	1	0.1601	0.0011	durchmischungst
1.3	145	1	11.1356	0.0768	runge_kutta
34.0	580	1	281.1590	0.4848	model
11.1	580	1	91.5683	0.1579	licht_und_tempe
0.0	19	1	0.3320	0.0175	remineralisatio
1.1	1	1	9.1443	9.1443	array_flash_vz
0.0	20	1	0.0300	0.0015	phosphat_pfeile
0.9	40	1	7.1805	0.1795	pfeil
3.9	36	1	32.0009	0.8889	output_grafik
0.2	2	1	1.5492	0.7746	eis_und_schnee
0.1	310	1	0.7073	0.0023	O2defizit
5.7	180	1	47.4092	0.2634	draw_arrays
18.0	71	1	148.4828	2.0913	o2_schraffierun
0.0	36	1	0.0197	0.0005	select_monat

Anlage 9: Farbfoto der Grafik des Computer-Bildschirms



L e b e n s l a u f

Name: Peter Gregor Hiering

Geburtsdatum: 1. Januar 1947

Geburtsort: München

Familienstand: nicht verheiratet

Bekenntnis: römisch-katholisch

Anschrift: Carl-Orff-Weg 10, 8025 Unterhaching

Schulausbildung: 4 Jahre Volksschule, 9 Jahre Gymnasium,
Reifeprüfung abgelegt im Juni 1966

Wehrdienst: 1966 - 1968: Zeitsoldat bei der Bundeswehr

Beruflicher Werdegang: 1969 - 1972: Studium des Lehramts an Volksschulen
an der Pädagogischen Hochschule München
1972 - 1980: Lehrer an Grund- und Hauptschulen
1972: Erste Staatsprüfung für das Lehramt an
Volksschulen
1975: Zweite Staatsprüfung für das Lehramt an
Volksschulen
1977: Berufung in das Beamtenverhältnis auf
Lebenszeit
1980 - 1984: Abordnung an das Staatsinstitut für
Schulpädagogik und Bildungsplanung (ISB)
seit Mai 1984: wissenschaftlicher Mitarbeiter am
Institut für die Didaktik der Biologie der Univer-
sität München
12.11.1985: Promotions-Vorprüfung

